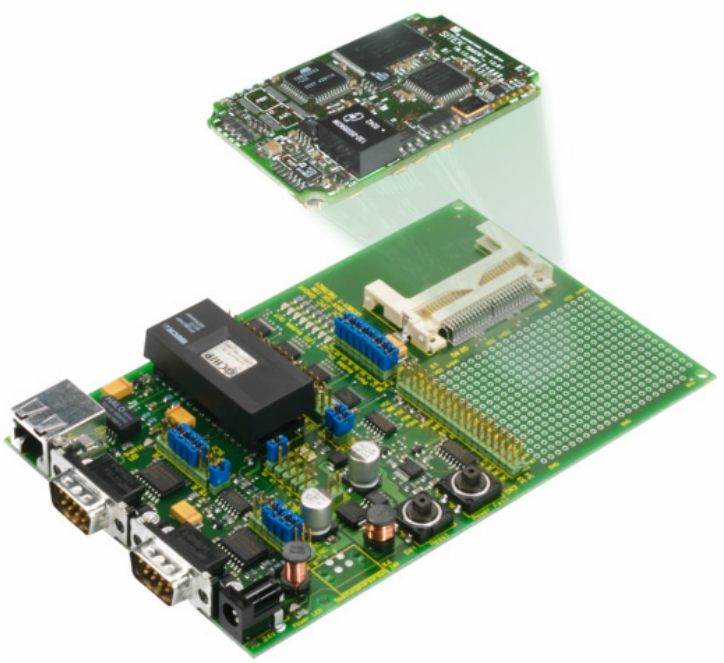
Staatliche Gewerbeschule für Energietechnik G 10
Fachschule für Technik

Facharbeit

Beschreibung, Inbetriebnahme und Anbindung eines embedded PCs



Prüfungsgebiet:	Automatisierungstechnik
Prüfer:	Herr Friedrich Geldmacher & Herr Matthias Pieper
Abgabetermin :	04.05.2006
Vorgelegt von:	Jens Hoyer & Sebastian Schoneberg

Inhaltsverzeichnis

1. Einleitung 1
2. Beschreibung des Systems 2
   2.1 Der Mikrocomputer - Die Alternative zum Mikrocontroller 2
   2.2 Das DK51 Board (Development Kit) 3
   2.3 Der IPC@CHIP SC13 der Firma Beck IPC GmbH 4
   2.4 Kommunikation mit dem IPC@CHIP 4
   2.5 Funktionen des IPC@CHIP 5
   2.6 Das Betriebssystem des IPC@CHIP 6
   2.7 Die Funktionsweise 7
3. Die Inbetriebnahme 7
   3.1 Chiptool 7
      3.1.1 IP Adresse 8
      3.1.2 Die Verbindungen zum Chip 9
   3.2 Total Comander und Telnet 11
   3.3 Testen ob der Webserver läuft 12
   3.4 Installation des Compact Flash 12
      3.4.1 Treiber für die Speicherkarte 12
      3.4.2 Erstellen einer Autoexec.bat 13
   3.5 TSP und RTS 14
4. Anbindung des DK51 16
   4.1 Das DK51 Board 16
   4.2 Das Modell des Mischautomaten 17
   4.3 Der Trafo 18
   4.4 Das Interface 18
5. Software 21
   5.1 Arbeiten mit CoDeSys 21
   5.2 Die Visualisierung 25
   5.3 Zielsystem starten 27
   5.4 Die Deklaration der Ein- und Ausgänge 27
   5.5 Starten des Projektes 28
6 Fazit 29
7 Literaturverzeichnis 29
8 Erklärung zur eigenständigen Anfertigung der Facharbeit 30
9 Einverständnisverklärung 30
10 Anhang 32
   10.1 Schaltplan der selbst angefertigten Platine 32
   10.2 Mischerprogramm und Visualisierung 33

Vorwort

Zum Abschluss der Ausbildung zum Techniker wird eine Facharbeit angefertigt. Unser erster Gedanke war, diese im Elektronikbereich zu erarbeiten. Als wir elektronische Bausätze analysierten, ist uns aufgefallen, dass selbst kleine elektronische Aufgaben mit Mikrocontrollern gelöst werden.

Wir wählten dann als Thema der Facharbeit den Mikrocontroller aus. Als wir Herrn Federrau befragten, empfahl er uns den ATMEGA 16/32 mit BASCOM zu programmieren. Herr Fangohr, der an der Schule einen Klassensatz Entwicklungsplatinen für den Mikrocontroller verwaltet, erklärte sich sofort bereit uns zu unterstützen. Als wir Herrn Tabel von unserer Facharbeit berichteten, empfahl er uns in C oder C + + zu programmieren und uns an Herrn Pieper zu wenden. Dieser unterstützte uns mit viel Engagement. Nachdem wir uns etwas in die ATMEGA Entwicklungsumgebung eingearbeitet hatten, schlugen uns Herr Geldmacher und Herr Pieper vor, die Facharbeit über einen Mikrocomputer zu schreiben. Der Chip ist in den genormten IEC Programmiersprachen, also auch in FUP (Funktionsplan), zu programmieren.

1. Einleitung

Das Ziel der Facharbeit ist mit der Entwicklungsumgebung DK51 ein Farbmischermodell aus dem SPS-Labor zu automatisieren. Die Facharbeit soll aufzeigen ob der Einsatz des DK51 Boards an der Schule sinnvoll ist. Die Facharbeit ist in vier Teile gegliedert. Die Beschreibung des Systems, die Inbetriebnahme und die Anbindung des DK51 Boards an die Hardware.

Der erste Teil beschreibt das Board und die Funktionen des IPC@CHIP SC13 der Firma Beck IPC GmbH und das dazugehörige Betriebssystem.

In dem zweiten Teil wird die Inbetriebnahme des DK51 Boards an den PC erläutert. Über verschiede Tools wird der IPC@CHIP SC13 eingerichtet, um eine Anbindung der CoDeSys Software zu ermöglichen.

Der dritte Teil befasst sich mit dem Anschließen des DK51 Boards an die weiteren Hardwarekomponenten und deren Funktionsbeschreibungen.

Im vierten Teil der Facharbeit wird das Schreiben von IEC Programmen und erstellen von Visualisierung mit CoDeSys beschrieben.

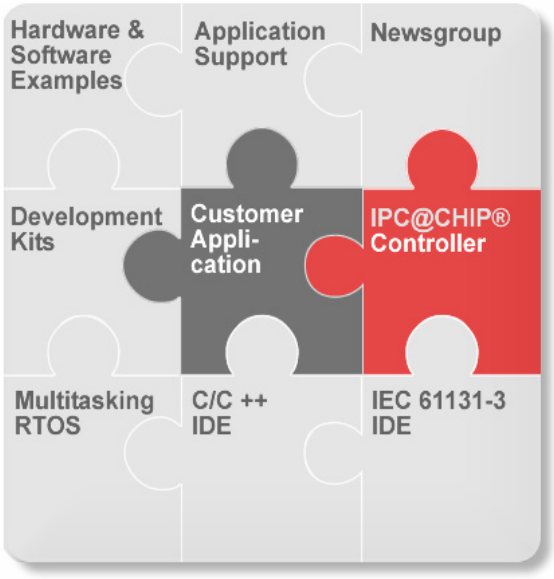
Am Ende der Facharbeit sind die Erfahrungen und die Einsatzmöglichkeiten des DK51 in einem Fazit zusammengefasst. Im Besonderen wird darauf eingegangen wie das System im Berufsschulunterricht eingesetzt werden kann.

2. Beschreibung des Systems

2.1 Der Mikrocomputer - Die Alternative zum Mikrocontroller

Bei herkömmlichen Mikrocontrollern kauft der Anwender den Controller, die Software, die Development Boards und das Betriebssystem von verschiedenen Lieferanten und führt die Installation und Anpassung der Software eigenständig durch. Bei technischen Schnittstellenproblemen zwischen Hardware und Software sind die Verantwortlichkeiten unklar und die technische Klärung ist mühsam und sehr zeitaufwändig. Die Lösung der IPC@CHIP Controller hingegen ermöglicht dem Anwender deutlich kürzere Entwicklungszeiten und niedrigere Entwicklungskosten als herkömmliche Mikrocontroller.[1]

Hinzuzufügen ist das der IPC@CHIP Controller ein Industrie PC ist. Es gibt wesentliche Unterschiede zwischen Office- und Industrie-Computern, die sich aus der Zuverlässigkeit und den erhöhten Anforderungen an die Beanspruchungen ergeben. Die Ausfallsicherheit und Zuverlässigkeit von Industrie-Computern hat einen anderen Stellenwert als die eines Office PC. Dieses wird durch hochwertige und toleranzarme Bauelemente und Komponenten erreicht. Die thermischen Beanspruchungen, die Resistenz gegen aggressive Gase, Spritzwasser und die mechanischen Belastungen werden durch entsprechende, Gehäuse- und Steckerdichtigkeit sowie engere Fertigungstoleranzen sichergestellt. Ein weiterer Aspekt ist der Lebenszyklus von Industrie-Computern. Die Lieferfähigkeit von Komponenten, CPUs und Modulen werden langjährig garantiert.[2]



Die Abbildung zeigt die nahtlose Integration der einzelnen IPC@CHIP Elemente zu einer Komplettlösung. Die durchschnittliche Entwicklungsdauer für die Kunden Applikationen liegt deshalb deutlich unter der Zeit die für den Einsatz eines Mikrocontrollers benötigt wird.[1]

Abb. 1
Quelle: www.beck-ipc.com/files/beck/infosheets/IS_DK51_DE_V3.pdf

[1] vgl. http://www.beck-ipc.com

[1] vgl. Computer & Automation 02/2006 S. 46ff.

[2] vgl. http://www.itwissen.info/de

2.2 Das DK51 Board (Development Kit)

Das Development Kit zum IPC@CHIP SC 13 hilft bei der Einarbeitung in den IPC@CHIP und ermöglicht den Aufbau von seriellen Prototypen. Auf der Abbildung 2 ist das DK51 Board mit einer kundenspezifischen Erweiterung zu sehen. Auf der Entwicklungsumgebung befindet sich der Chip SC13.

Es handelt sich hier um einen Embedded Controller. Er bietet das bewährte RTOS Real Time Operating System an. Auf dem Board befinden sich z. B. eine Ethernet Schnittstelle, zwei serielle Schnittstellen, ein Daten- und ein Adressbus. Der Chip SC13 kann in Hochsprache oder mit CoDeSys@CHIP IEC61131 programmiert werden.[1]

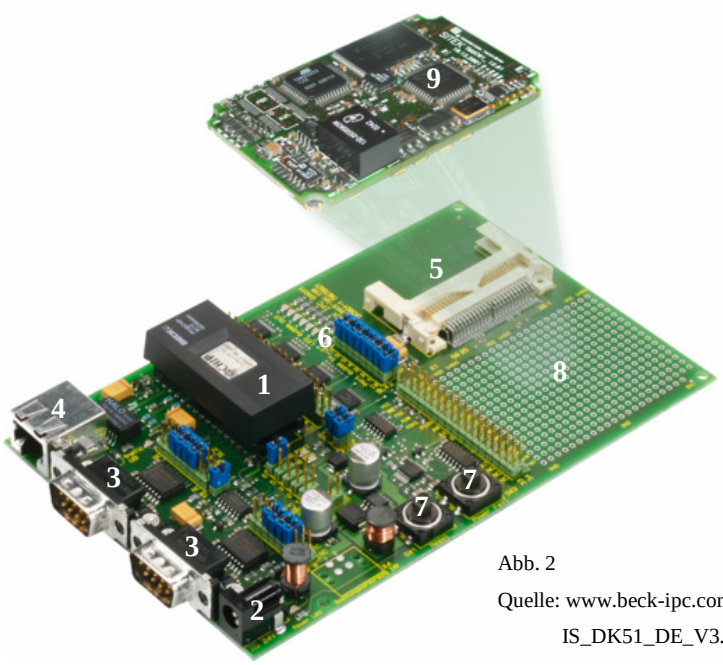


Abb. 2
Quelle: www.beck-ipc.com/files/beck/infosheets/IS_DK51_DE_V3.pdf

(1) IPC@CHIP SC13
(2) Spannungsversorgung (24 VDC Eingang)
(3) Zwei RS232 Schnittstellen (COM und EXT)
(4) 10/100MBit/s Ethernet Schnittstelle
(5) Compact Flash Schnittstelle
(6) 16 programmierbare I/O Pins mit LED Anzeige
(7) Power Fail und Reset Taster
(8) Lochrasterfeld für kundenspezifische Erweiterungen
(9) Kundenspezifische Erweiterung

[1] vgl. http://www.beck-ipc.com

2.3 Der IPC@CHIP SC13 der Firma Beck IPC GmbH

Abb. 3
Quelle: www.beck-ipc.com

Bei dem verwendeten IPC@CHIP SC13 der Firma Beck IPC GmbH handelt es sich um einen Embedded Web Controller. In einem Embedded System sind Hard- und Software kombiniert und aufeinander abgestimmt, um eine spezielle Funktion zu erfüllen. Das verarbeitende Teilsystem arbeitet nicht als Stand-alone-Lösung, sondern ist in dem überwachenden bzw. steuernden System integriert (embedded). Ein Embedded Web Controller ist ein „Steuerungssystem“, das für die Web-Kommunikation entwickelt wurde. Dieser Chip ist LAN-fähig und stellt einen Web-Server zur Verfügung. Daher eignet er sich sehr gut für Projekte, bei denen die Steuerung oder Überwachung über Inter- oder Ethernet stattfindet. Dank seiner umfangreichen Ausstattung mit Industrie- und IT- Schnittstellen sowie zahlreichen Kommunikationsprotokollen lassen sich mit dem Chip vielfältige Kommunikationsaufgaben realisieren.[1]

2.4 Kommunikation mit dem IPC@CHIP

Das RTOS des IPC@CHIP enthält unter anderem einen FTP-Server, einen HTTP-Server und einen Telnet-Server. Mit einem FTP-Client können die programmierten Applikationen in den Speicher des Chips kopiert werden. FTP (file transfer protocol) ist ein Übertragungsprotokoll, das im weltweiten Internet oder in lokalen Netzen eingesetzt wird, die TCP/IP als Netzwerkprotokoll benutzen. Über den Telnet-Client kann ein Command-Shell geöffnet werden um dort Kommandozeilenbefehle einzugeben, also RPC (Remote Process Controlling) durchzuführen. Der Unterschied zwischen dem Protokoll für Telnet und z. B. dem FTP ist, dass nicht nur auf die Dateien von einem entfernten Rechner zugegriffen werden, sondern auf diesem Host z. B. auch Applikationen starten kann.
Zusätzlich besteht die Möglichkeit, über die seriellen Schnittstellen des IPC@CHIP Daten auszutauschen. Durch das Zufügen der entsprechenden Treiberbausteine wird die Schnittstelle Wahlweise als RS232, RS422 oder RS485 ausgeführt. Das im RTOS integrierte PPP Protokoll (Server und Client) ermöglicht TCP/IP Verbindungen auch über das Festnetz- und GSM/GPRS-Modem ins Internet oder ins Firmen-Intranet.[2]

---

[1] vgl. http://www.beck-ipc.com

[1] vgl. Elektronik Journal 04/2002 S.80ff.

[2] vgl. K-Magazin 01/2004 S. 36

[2] vgl. http://www.beck-ipc.com

2.5 Funktionen des IPC@CHIP

Der IPC@CHIP besitzt einen multiplexed 8-Bit-Daten und Adressbus, d. h. einen Bus der gleichzeitig Adress- und Datenbus ist. Dieser wird auf dem DK51 in separate Adress- und Datenbusse getrennt (demultiplexed). Weitere Funktionen sind die internen Timer, die über die Eingänge TIMERIN getriggert werden können und auf die Ausgänge TIMEROUT das entsprechende Ausgangssignal ausgeben. Außerdem werden ein I²C- und ein SPI-Bus vom IPC@CHIP zur Verfügung gestellt. Für diese Facharbeit werden die In- und Outputs (PIO) genutzt. Die Ein und Ausgänge (PIO) sind von der Funktion ähnlich wie bei der SPS. Die Spannung beträgt 5V (TTL Pegel). Der Strom je Ausgang darf max. 20mA betragen, für alle Ausgänge zusammen jedoch nicht mehr als 60mA. Da die meisten Pins Mehrfachbelegungen haben, müssen vor ihrer Verwendung diese entsprechend initialisiert werden. In der unteren Abbildung ist die Pinbelegung des IPC@CHIP zu sehen.[1]

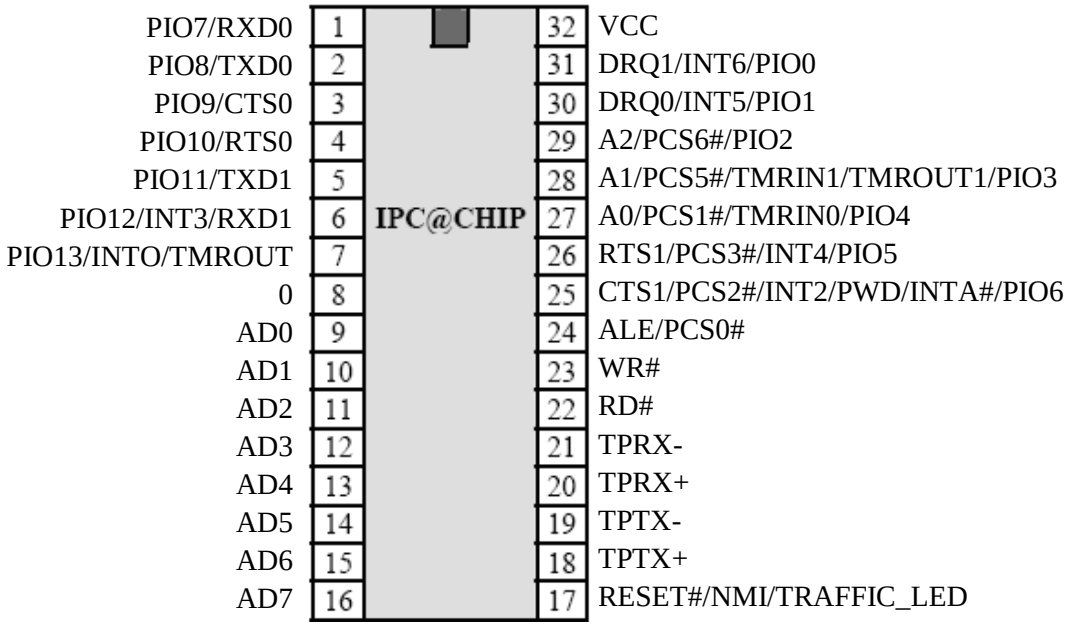


Abb. 4

Quelle: www.beck-ipc.com

Wissenswertes

Der IPC@CHIP kann ca. 8,75 Millionen Befehle pro Sekunde verarbeiten. Dieser Wert wurde mit dem Dhrystone Benchmarktest ermittelt. Dieser Chip wurde bereits in den verschiedensten Bereichen verwendet, z. B. zur Steuerung eines Roboters, für eine Temperaturregelung und für eine Zugangskontrollsteuerung.[2]

---

[1] vgl. K-Magazin 01/2004 S. 36

[1] vgl. Mark t& Technik 16/04/2004 S. 12

[1] vgl. http://www.beck-ipc.com

[2] vgl. http://www.beck-ipc.com

2.6 Das Betriebssystem des IPC@CHIP

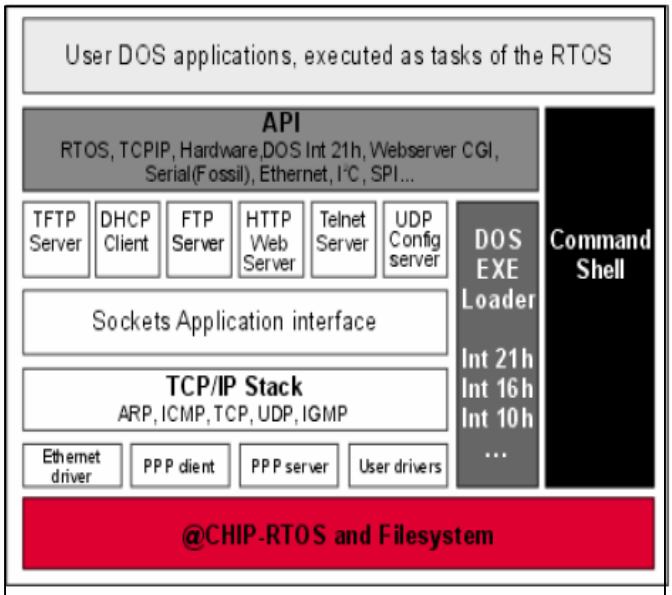


Abb. 5

Quelle: www.beck-ipc.com

Das Betriebssystem des IPC@CHIP ist ein multitaskingfähiges Real Time Operating System (RTOS), auf dem 35 Tasks (davon bis zu 12 Dos-Applikationen) und bis zu 60 Timer, Semaphore und Event Groups ausgeführt werden können. Außerdem stellt das Betriebssystem einen FTP- und einen Telnet-Server und ein Application Programmers Interface (API) als Schnittstelle zur Verfügung. Ein Echtzeitsystem ist gekennzeichnet durch ein voraussagbares (deterministisches) Zeitverhalten. Wird also etwa einem Echtzeitsystem die Aufgabe erteilt (durch Konfiguration), bestimmte Routinen innerhalb eines vorgegebenen Zeitrasters auszuführen, so wird dies innerhalb vorgegebener zeitlicher Toleranzen passieren, ansonsten wird das als Versagen des Gesamtsystems angesehen. Auf eine Steuerung, im Sinne von CoDeSys bezogen, heißt das, dass ein Task innerhalb gegebener (vorher bekannter) Toleranzen aufgerufen wird.

Die Erfahrung hat gezeigt, dass diese Grenzen von Windows NT nicht eingehalten werden, NT ist also kein richtiges Echtzeitbetriebssystem. Die Hardware in einem PC ist aber so beschaffen, dass es mit Mitteln der Software möglich ist, ein verlässliches Task-Scheduling zustande zu bringen. Die Echtzeiterweiterung von Windows NT ist z. B. ein NT-Treiber, der eine Interrupt-Service-Routine installiert, die zyklisch vom Timertick der PC-Hardware aufgerufen wird.

Diese Routine übernimmt nun die Aufgabe, von CoDeSys definierte Tasks anzusprechen und/oder die Ausführung des Betriebssystems weiter oder wieder zu zulassen.[1]

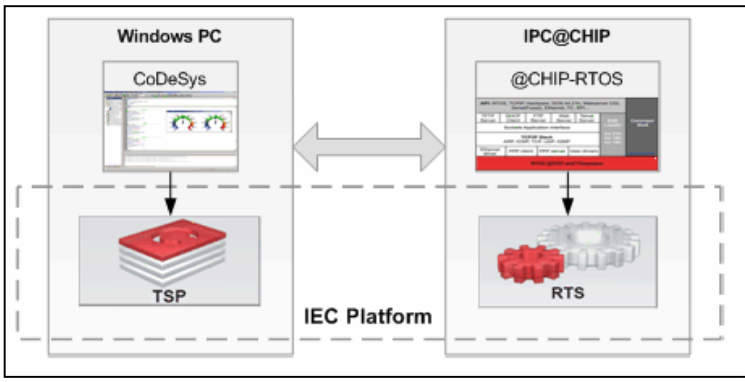
[1] vgl. K-Magazin 01/2004 S. 36

[1] vgl. http://www.beck-ipc.com

[1] vgl. Computer & Automation 02/2006 S. 46ff.

2.7 Die Funktionsweise

Das Programm CoDeSys, auf einem Windows PC, ist ein Hardware unabhängiges Programm und beinhaltet die genormten IEC Programmiersprachen (Strukturierter Text, Anweisungsliste, Kontaktplan, Ablaufsteuerung und Funktionsplan). Das TSP (Target Support Package) wird in CoDeSys eingefügt und ermöglicht den Datenaustausch von CoDeSys zum RTS (Run Time System). Auf dem IPC@CHIP wird die myrts Datei eingefügt und sorgt für die Kommunikation zwischen dem TSP und dem CHIP. Das Target Support Package und das Run Time System dienen als Schnittstelle zwischen der Software (CoDeSys) und der Hardware (DK51).[1]



Die Kommunikationshierarchie zwischen dem PC und dem IPC@CHIP SC13.

Abb. 6
Quelle: www.beck-ipc.com

3. Inbetriebnahme des DK51

3.1 Chiptool

Auf der Homepage der Firma Beck kann das Programm Chiptool herunter geladen werden (http://www.beck-ipc.com). Steht kein DHCP Dienst zur Verfügung wird das Programm benötigt um die IP Adresse des Chips zu beziehen. Über Chiptool können dann verschiedene Verbindungen zum Chip hergestellt werden (FTP Verbindung, Telnet Verbindung und Serielle Verbindung). Über diese Verbindungen können Dateien auf dem Chip gespeichert, gelöscht oder ausgeführt werden. Dieses kann dann über eine Serielle oder Ethernet Schnittstelle erfolgen.[2]

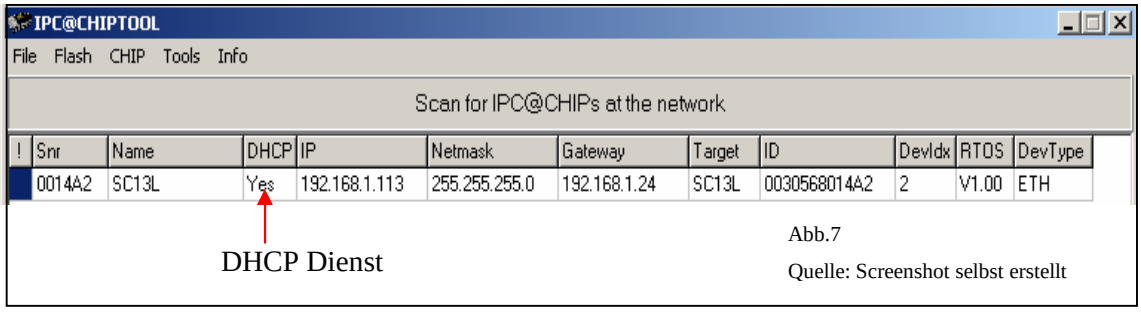

IPC@CHIPTOOL
File Flash CHIP Tools Info
Scan for IPC@CHIPs at the network
! Snr Name DHCP IP Netmask Gateway Target ID DevIdx RTOS DevType
0014A2 SC13L Yes 192.168.1.113 255.255.255.0 192.168.1.24 SC13L 0030568014A2 2 V1.00 ETH
DHCP Dienst

Abb.7
Quelle: Screenshot selbst erstellt

[1] vgl. http://www.beck-ipc.com

[2] vgl. IPC World 02/2006 S. 30ff.

3.1.1 IP Adresse

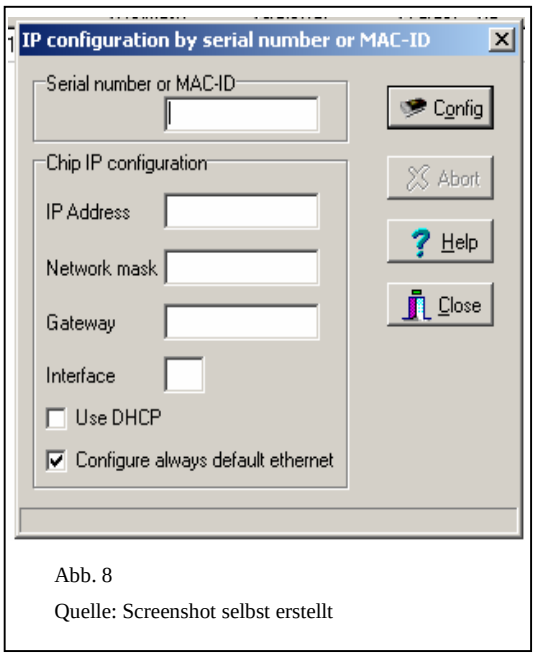


Abb. 8
Quelle: Screenshot selbst erstellt

Wenn kein DHCP Server vorhanden ist der an den Chip automatisch eine IP Adresse vergibt, muss dies manuell mit dem Programm Chiptool geschehen. Nachdem das Programm installiert und gestartet worden ist, muss geprüft werden, ob ein DHCP Server vorhanden ist. Ist dies der Fall, bekommt der Chip automatisch seine IP Adresse zu gewiesen. Ist dieser nicht vorhanden muss die IP Adresse manuell eingestellt werden. Das Menü zur manuellen Einstellung der IP Adresse kann über ***CHIP*** à ***IP Config***. gestartet werden. Die Mac-ID wird von der vorherigen Seite übernommen. Bei Chip IP Configuration wird die gewünschte IP Adresse eingegeben. Die Network Mask hängt von dem Netzwerk ab (Klasse C Network Mask = 255.255.255.0, Klasse B Network Mask = 255.255.0.0). Die Einstellungen bei Network Mask und Gateway müssen mit den PC übereinstimmen. Auch die IP Adresse kann nicht frei gewählt werden, die ersten Zahlen müssen mit denen am Rechner übereinstimmen. Die IP Adresse, die Network Mask und das Gateway des PC kann in der Windows Eingabeaufforderung mit IP Config. ausgelesen werden. Hier sind jetzt die Einstellungen des Rechners sichtbar (untere Abb.).[1]

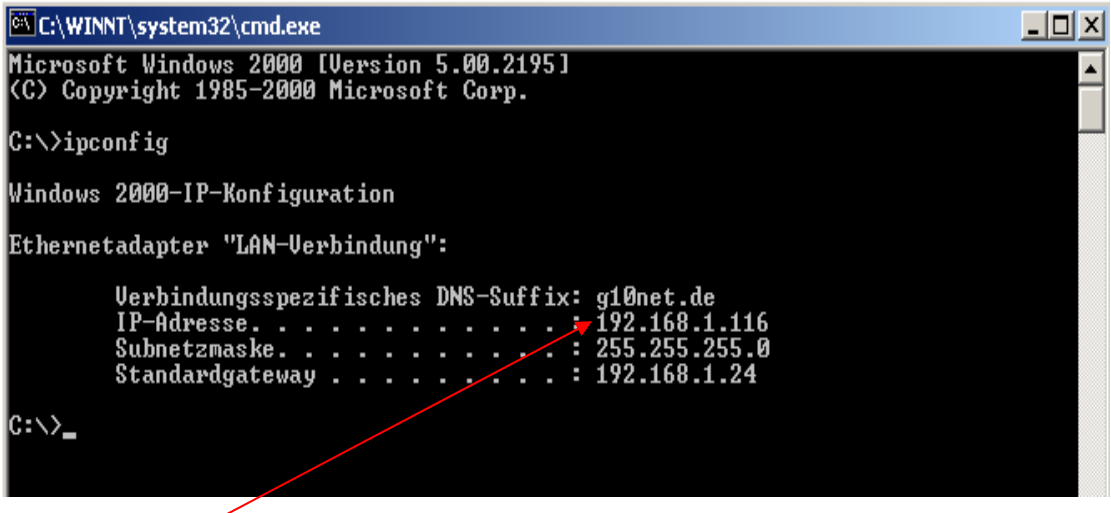

C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>ipconfig
Windows 2000-IP-Konfiguration
Ethernetadapter "LAN-Verbindung":
Verbindungsspezifisches DNS-Suffix: g10net.de
IP-Adresse. . . . . : 192.168.1.116
Subnetzmaske. . . . . : 255.255.255.0
Standardgateway . . . . . : 192.168.1.24
C:\>_
z. B. IP Adresse des PC

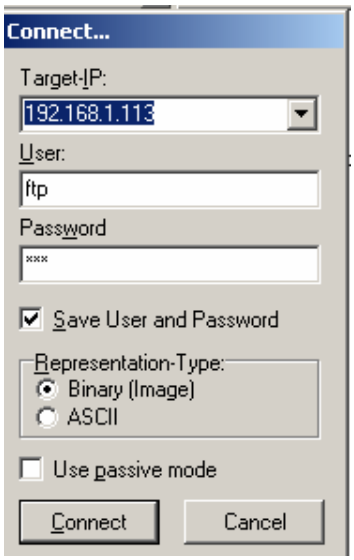
Abb. 9
Quelle: Screenshot selbst erstellt

Die IP Adresse des Chips muss 192.168.1.xxx lauten, weil es sich bei dem Netzwerk um ein Klasse C Netzwerk handelt (siehe Subnetzmaske).

[1] vgl. http://www.fibel.org/linux/lfo-0.6.0/node469.html

3.1.2 Die Verbindungen zum Chip[1]

FTP Verbindung

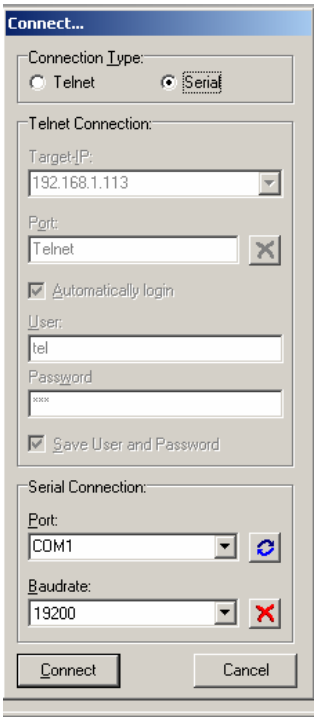


Das Connect Fenster wird über ***Tools à FTP-Client*** geöffnet. Bei der Taget IP wird die IP Adresse des Chips eingetragen (z.B. 192.168.1.113) bei User und Passwort wird jeweils ***„FTP“*** eingegeben. Der Haken bei „Save User and Passwort“ speichert die Einstellungen. Bei Betätigung des Connect Butten wird die FTP Verbindung hergestellt. Dateien die auf dem Chip sind, werden angezeigt und können kopiert oder gelöscht werden. Das Ausführen einer Datei mit der FTP Verbindung ist nicht möglich. Um zum Beispiel den Inhalt der chip.ini Datei zu lesen, muss die Datei mit der FTP Verbindung auf den Rechner kopiert werden. Hier kann die Datei mit dem Editor geöffnet werden.

Abb. 10

Quelle: Screenshot selbst erstellt

Serielle Verbindung



Wenn kein Netzwerk vorhanden ist und auch kein Crosslinkkabel zur Verfügung steht, kann mit Hilfe von Chiptool auch eine serielle Verbindung über eine RS 232 Schnittstelle zum Chip hergestellt werden.

Um in das Menü zum Herstellen einer seriellen Verbindung zu kommen, ist wie in das Menu zum Herstellen einer Telnet Verbindung vorzugehen **(*Tools à Terminal*).**

Als Connection Type ist hier ***Serial*** auszuwählen. Die weiteren Einstellungen sind identisch mit der Telnet Verbindung. Durch betätigen der ***Connect*** Taste werden die Einstellungen übernommen.

Abb. 11

Quelle: Screenshot selbst

[1] Getting Started V2.3 S. 31ff.

Telnet Verbindung

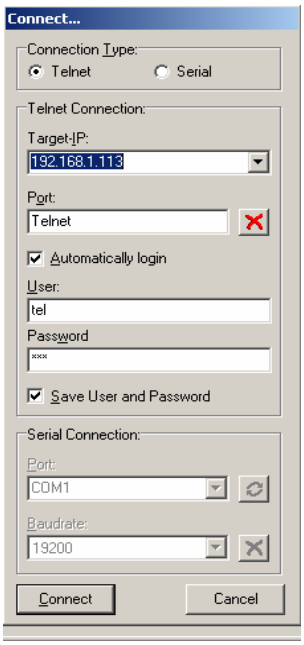


Abb. 12

Quelle: Screenshot selbst erstellt

Die Telnet Verbindung ermöglicht das Ausführen von Dateien auf dem Chip.

Um die Verbindung herzustellen, muss das Connect Fenster über ***Tools*** à ***Terminal*** geöffnet werden. Als Connection Type ***Telnet*** auswählen und bei Target IP die Adresse des Chips eintragen. Bei User und Passwort muss jeweils **„tel“** eingetragen werden.

Wenn jetzt bei „Save User and Passwort“ ein Haken gesetzt wird, werden die Einstellungen gespeichert und müssen nicht bei einen Neustart erneut eingegeben werden.

Im unteren Bild wird z. B. durch den Befehl ***dir*** alle Dateien des Laufwerkes A angezeigt. Die Datei extide.exe auf dem Chip, kann dann z. B. mittels der Eingabetaste ausgeführt werden.

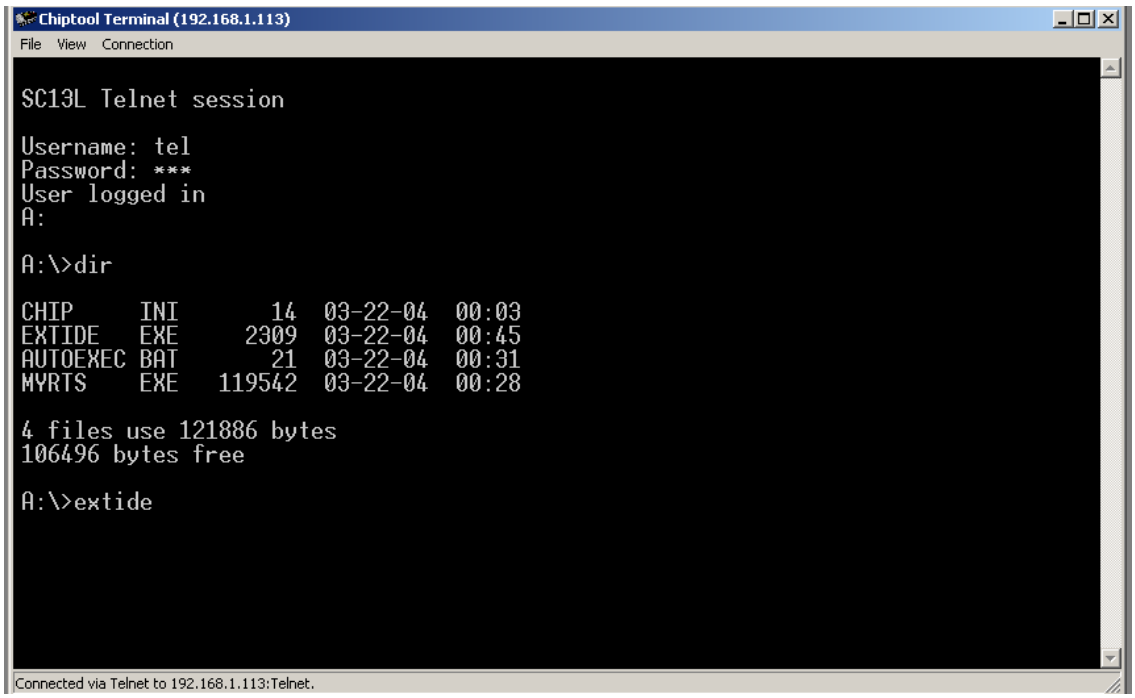


Abb. 13

Quelle: Screenshot selbst erstellt

3.2 Total Comander und Telnet

Wenn keine aktuelle Version des Programms Chiptool zur Verfügung steht, muss die FTP Verbindung über den Win Comander und die Telnet Verbindung über die Windows Eingabeaufforderung hergestellt werden. Win Comander wird zusammen mit dem DK51 System mitgeliefert. Soll eine FTP Verbindung hergestellt werden, muss der Win Comander installiert, das Programm ausgeführt und über ***Befehle à FTP neue Verbindung*** eine FTP Verbindung hergestellt werden.

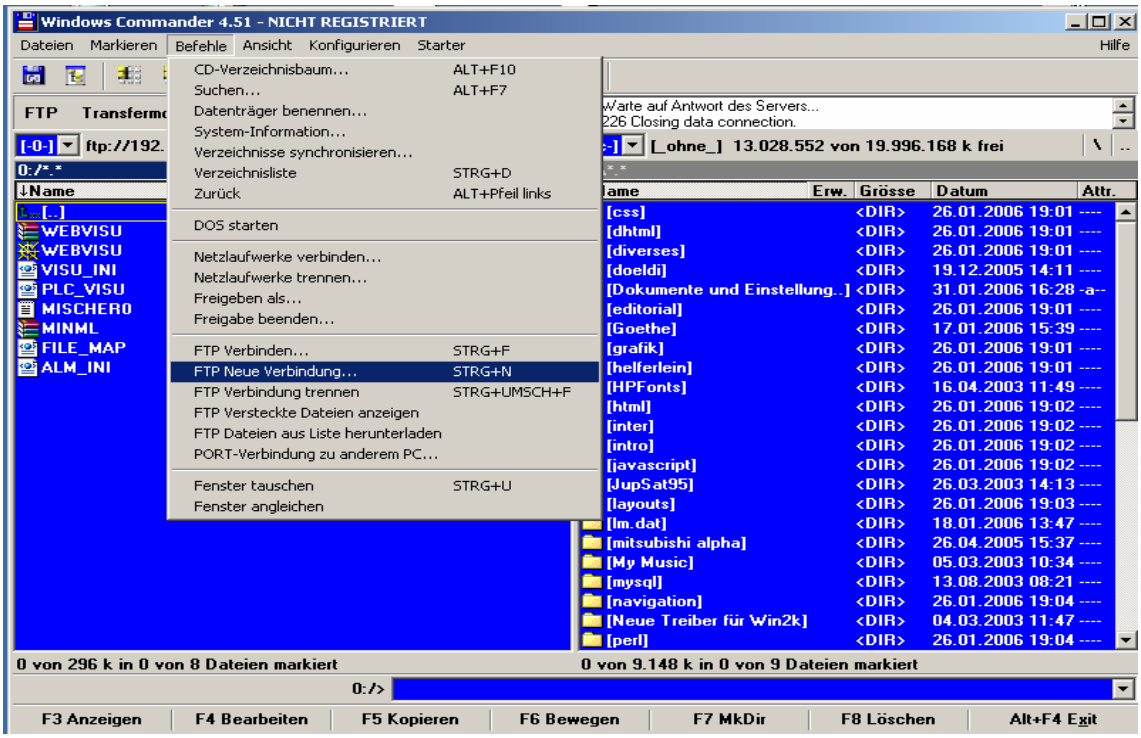

Windows Commander 4.51 - NICHT REGISTRIERT
Dateien Markieren Befehle Ansicht Konfigurieren Starter Hilfe
CD-Verzeichnisbaum... ALT+F10
Suchen... ALT+F7
Datenträger benennen...
System-Information...
Verzeichnisse synchronisieren...
Verzeichnisliste STRG+D
Zurück ALT+Pfeil links
DOS starten
Netzlaufwerke verbinden...
Netzlaufwerke trennen...
Freigeben als...
Freigabe beenden...
FTP Verbinden... STRG+F
FTP Neue Verbindung... STRG+N
FTP Verbindung trennen STRG+UMSCH+F
FTP Versteckte Dateien anzeigen
FTP Dateien aus Liste herunterladen
PORT-Verbindung zu anderem PC...
Fenster tauschen STRG+U
Fenster angleichen
Warte auf Antwort des Servers...
226 Closing data connection.
[_ohne_] 13.028.552 von 19.996.168 k frei
0 von 296 k in 0 von 8 Dateien markiert
0 von 9.148 k in 0 von 9 Dateien markiert
F3 Anzeigen F4 Bearbeiten F5 Kopieren F6 Bewegen F7 MkDir F8 Löschen Alt+F4 Exit

Abb.14

Quelle: Screenshot selbst erstellt

Um eine Telnet Verbindung herzustellen, muss die Windows Eingabeaufforderung gestartet werden. Den Befehl ***Telnet*** eingeben, im nächsten Feld den Befehl ***open,*** dann die IP Adresse des Chips. Im nächsten Fenster wird bei user und password jeweils ***tel*** eingeben.

Im nächsten Schritt wird das Laufwerk ausgewählt, und die gewünschte Datei auf dem Chip gestartet. Befehle müssen in der Eingabeaufforderung jeweils mit der Eingabetaste bestätigt werden (einfacher ist es mit einer aktuellen Version von Chiptool zu arbeiten).

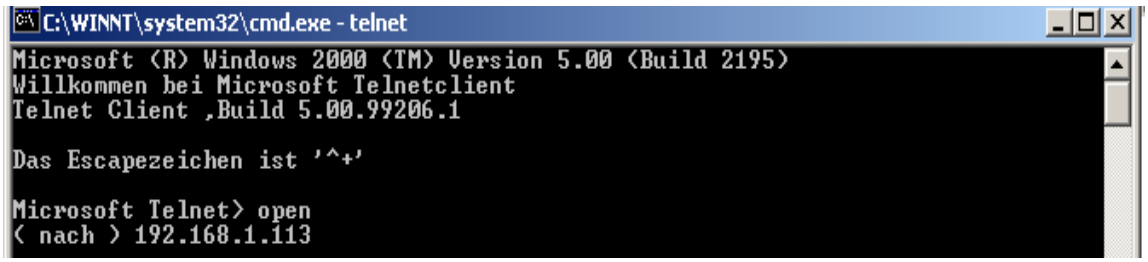

C:\WINNT\system32\cmd.exe - telnet
Microsoft (R) Windows 2000 (TM) Version 5.00 (Build 2195)
Willkommen bei Microsoft Telnetclient
Telnet Client ,Build 5.00.99206.1
Das Escapezeichen ist '^+'
Microsoft Telnet> open
(nach) 192.168.1.113

Abb.15

Quelle: Screenshot selbst erstellt

3.3 Testen ob der Webserver läuft

Auf dem Chip läuft im Hintergrund ein Webserver. Die Funktion kann zum Beispiel über den Microsoft Internet Explorer überprüft werden. In die Adresszeile ***„http://IP-Adresse des Chips“*** eingegeben, die aktuelle Chip Konfiguration wird angezeigt. Der Speicher des Chips muss mit einer Compact Flash erweitert werden um Visualisierungen auf dem Chip zu spielen.

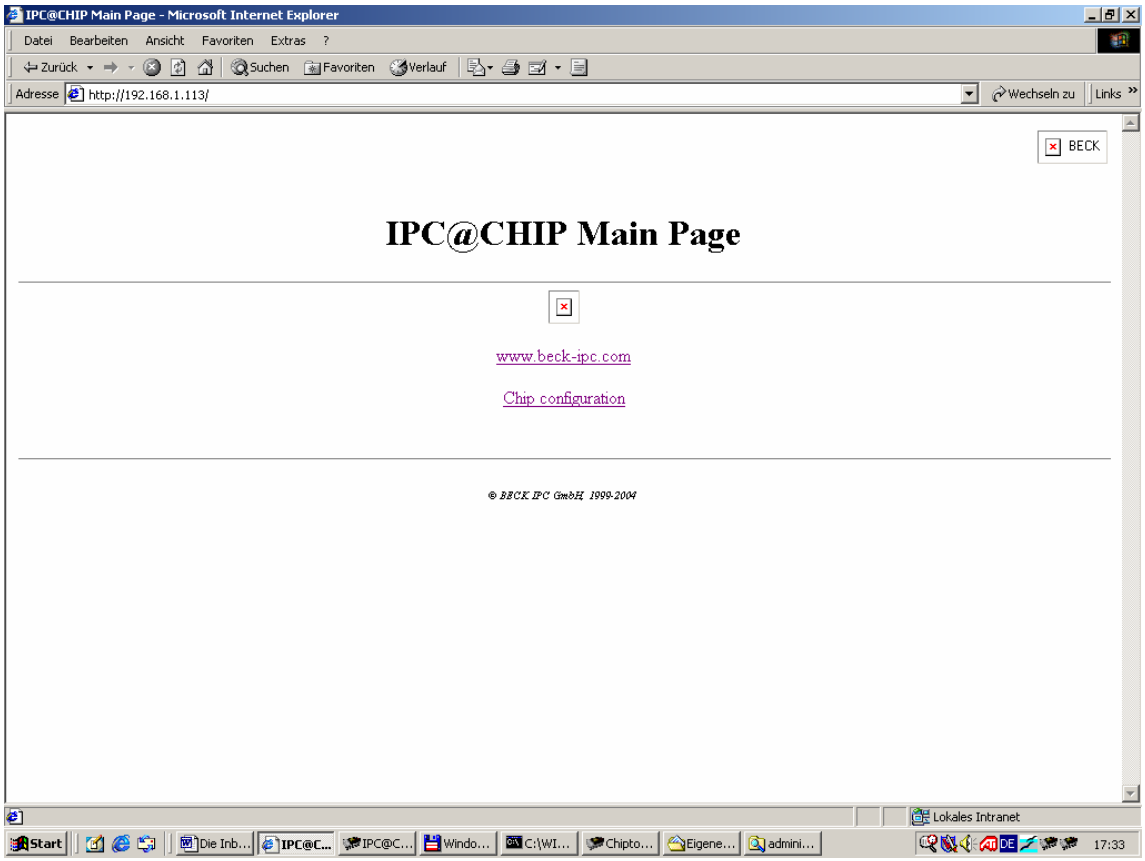


Abb.16

Quelle: Screenshot selbst erstellt

3.4 Installation des Compact Flash[1]

3.4.1 Treiber für die Speicherkarte

Der Treiber (EXTIDE) für die Speicherkarte kann von der Homepage der Firma Beck herunter geladen werden (www.Beck-ipc.com). Die readme.txt Datei gibt Auskunft über die richtige Version (paradigm 20 Chip). Nach der Auswahl der extide.exe Datei muss diese mit Hilfe einer FTP Verbindung auf den Chip kopiert werden. Über die Telnet Verbindung wird diese dann ausgeführt. Die Karte wird jetzt als zweites Laufwerk angezeigt. Wenn auf die Speicherkarte nicht zugegriffen werden kann, muss diese noch partitioniert werden. Dafür eignet sich das Programm Fdisk, es kann eine oder mehrere Partitionen auf der Karte erstellen.

[1] Getting Started V2.3 S. 45

3.4.2 Erstellen einer Autoexec.bat

Bei einer Spannungsunterbrechung oder einem Reset muss die Speichererweiterung die zur Verfügung steht über die Telnet Verbindung neu gestartet werden. Bei Erstellung einer Autoexec.bat Datei startet diese alle Dateien automatisch. Die Datei wird mit dem Editor erstellt und muss Autoexec.bat heißen. Im Editor müssen die Dateien eingefügt werden, die automatisch gestartet werden sollen(z. B. extide.exe). Danach speichert man diese Datei ab, und kopiert die Datei mit der FTP Verbindung auf den Chip. Nach jedem Neustart oder Reset werden diese Dateien jetzt automatisch nacheinander gestartet.

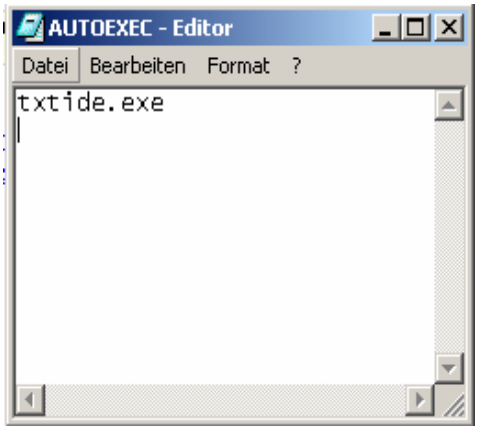


Abb.17
Quelle: Screenshot selbst erstellt

Vorbereitung der Programmierung mit CoDeSys

Nachdem der Chip erfolgreich konfiguriert wurde, muss eine Programmiersprache ausgewählt werden. Das Programm kann in Hochsprache oder mit der CoDeSys@CHIP IEC61131 Software programmiert werden. CoDeSys steht für Controller Development System. Es ist eine Entwicklungsumgebung für Steuerungen. CoDeSys ermöglicht einen einfachen Einstieg in die IEC Programmierung. Die Benutzung der Editoren und der Debugging-Funktionen hat CoDeSys aus Entwicklungsumgebungen höherer Programmiersprachen zum Vorbild (wie etwa Visual C++). Das Programm für die Ansteuerung des Mischermodells ist mit der CoDeSys Software in FUP erstellt.[1]

Als Schnittstelle zwischen der Software CoDeSys und dem Chip dienen zwei Dateien. Die myrts.exe Datei muss auf den Mikrocomputer gestartet sein. In CoDeSys muss die TSP Datei einmal ausgeführt werden. Bevor die Dateien erstellt oder aus dem Internet herunter geladen werden, muss CoDeSys auf dem Computer installiert werden.

[1] vgl. Computer & Automation 02/2006 S. 46ff.

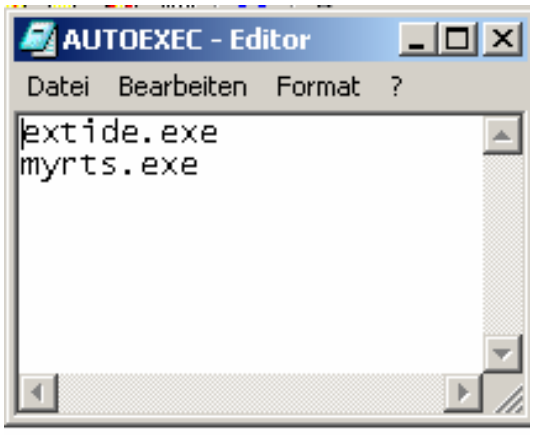
3.5 TSP und RTS[1]

Nachdem CoDeSys auf dem Rechner installiert ist, müssen die Dateien (TSP und RTS) entweder selbst erstellt oder aus dem Internet herunter geladen werden.

Mit dem Programm IEC Plattform Builder auf der getting start CD können beide Dateien nach individuellen Ansprüchen erstellt werden. Einfacher ist es jedoch, die beiden Dateien aus dem Internet von der Homepage der Firma Beck herunter zu laden. Es ist trotzdem sinnvoll den IEC Plattform Builder zu installieren.

Mit der Hilfe dieses Programms können die Standard Einstellungen der Dateien angezeigt werden. Nach dem herunterladen, muss die readme Datei geöffnet werden. Hier stehen wichtige Informationen für die Datei Auswahl. Um Programme mit Web Visualisierungen auf den Chip zu speichern, muss der Ordner DK51WV ausgewählt werden. In diesem Ordner befinden sich die Dateien myrts.exe und TSP.bat. Die Datei myrts.exe muss mit Hilfe einer FTP Verbindung auf den Chip kopiert, und mit einer Telnet Verbindung ausgeführt werden. Da die Datei nach jedem Neustart des Chips immer wieder ausgeführt werden muss, ist es sinnvoll, diese in die bereits vorhandene Autoexec.bat zu kopieren. Das erstellen einer Autoexec.bat Datei wurde im Kapitel 3.4.2 ausführlich beschrieben. Die Dateien starten in der Reihenfolge, in der sie in der Autoexec.bat stehen.

Nachdem die Datei TSP.bat ausgeführt wurde, muss in CoDeSys unter dem Register ***Ressourcen à Zieleinstellungen*** (unten links) das DK51WV System ausgewählt werden. Bei der Erstellung eines neuen Projektes sollte die Zieleinstellung am Anfang erfolgen. *Mit* ***Datei à Neu*** wird ein neues Projekt erstellt, es öffnet sich das Fenster Zielsystem Einstellungen, hier kann direkt das ***DK51WV*** ausgewählt werden.



Die extide.exe muss vor der myrts.exe ausgeführt werden, da die extide.exe der Treiber für die Speicherkarte ist. Wenn die myrts.exe gestartet wird, muss bereits der gesamte Speicherplatz zur Verfügung stehen.

Abb. 18

Quelle: Screenshot selbst erstellt

[1] Getting Started V2.3 S. 29

[1] Vgl. Erste Schritte mit CoDeSys

Weitere Einstellungen in der CoDeSys Software

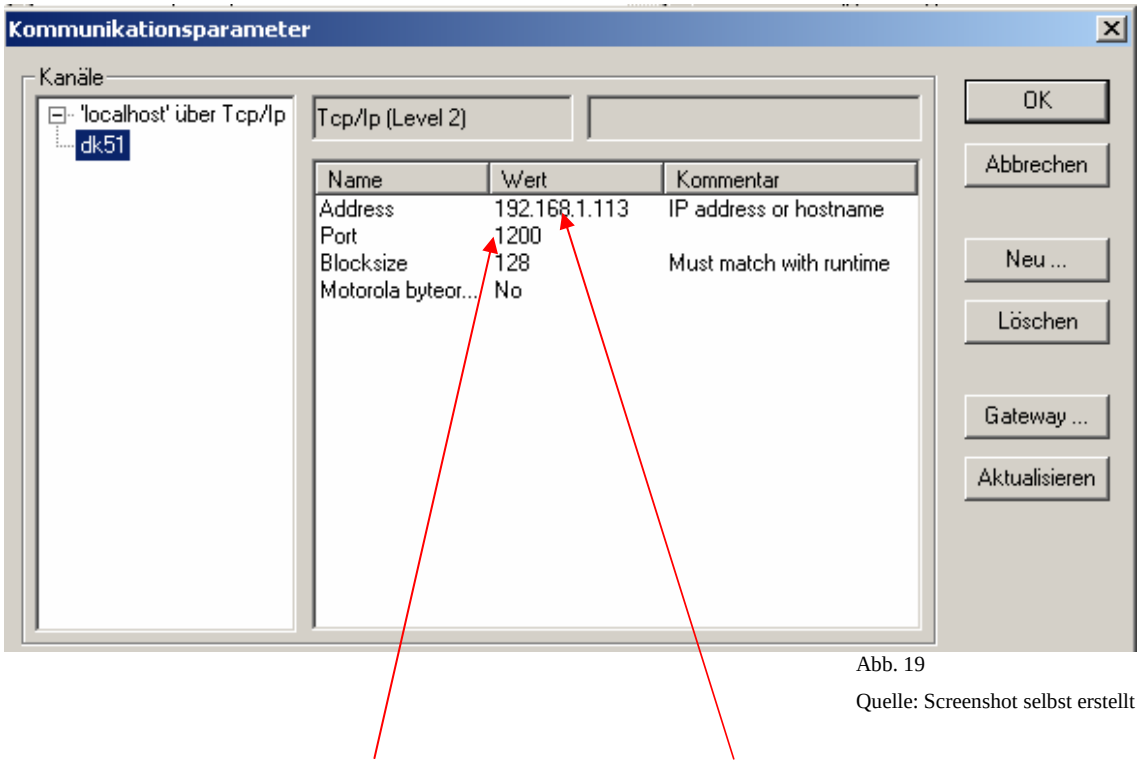
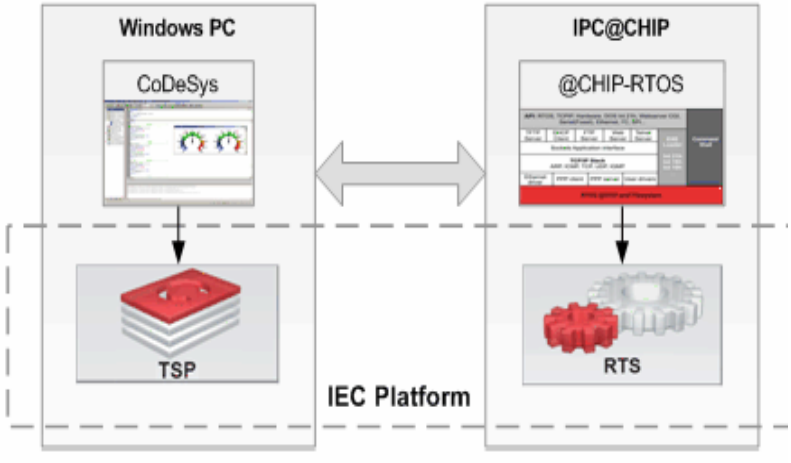


Abb. 19

Quelle: Screenshot selbst erstellt

Wichtig: Der Port muss auf 1200 stehen. IP Adresse des Chips

Über ***Online à Kommunikationsparameter*** können die Einstellungen überprüft werden (der Port 1200 und die richtige IP Adresse des Chips).



Die Inbetriebnahme und die Konfiguration des IPC@Chips ist abgeschlossen und es kann mit der Programmierung begonnen werden.

Abb. 20

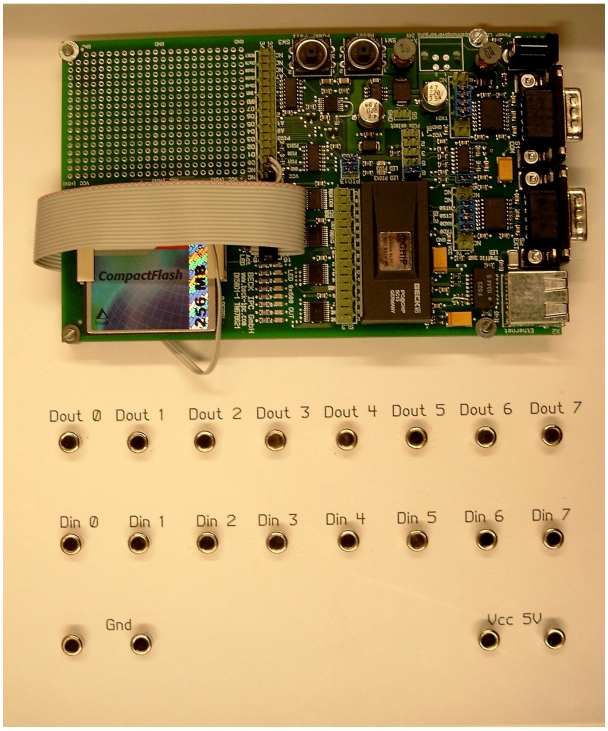
Quelle: www.beck-ipc.com

4. Anbindung des DK51

Um das DK51 zu testen soll im Rahmen der Facharbeit das Mischermodell der G10 gesteuert werden. Da dieses Modell mit 24V betrieben wird muss ein Interface angefertigt werden.

4.1 Das DK51 Board

Das Development Kit arbeitet mit einem 5V TTL Ausgangssignal. Das SPS Modell mit dem der Mischautomat simuliert wird, benötigt ein SPS Signal von 24 Volt. Um dieses Problem zu beheben wurde ein Interface entwickelt. Dieses wandelt die 5V TTL Ausgangssignale vom DK51 in 24V SPS Signale um. Die 24V vom SPS Modell werden in ein 5V TTL Signal umgewandelt, um die Eingänge des DK51 Boards anzusteuern.
Das Interface wird mit 5V vom DK51 und mit 24V vom Trafo versorgt. Unsere Hardware besteht somit aus dem DK51 Board, dem Mischermodell, dem Interface und einem 24V Trafo. Alle Hardwarekomponenten sind jeweils auf einer weißen Grundplatte befestigt. Diese ist dann für den Einsatz am Laborrahmen geeignet. Zusätzlich wurden alle zu verarbeitenden Signale auf Buchsen geführt. Somit sind spätere Änderungen an der Schaltung mit Laborleitungen schnell und unkompliziert möglich. An der Steckerleiste SL4 stehen die 8 Digitalen Ein- und Ausgänge zur Verfügung. Diese sind mit einem 16 adrigem Flachbandkabel auf die Buchsen Din0 bis Din7 und Dout0 bis Dout7 geführt. Die Buchsen VCC 5V und Gnd werden von der Steckerleiste SL5 gespeist, zusätzlich wird über ein zweipoliges Kabel 5V für das Interface bereitgestellt.



Stückliste

1 Grundplatte (weiß)
4 Gewindeschrauben
20 Buchsen in Silber
1 Stück Flachbandkabel
1 Steckerleiste (16 polig)
1 Stück Kabel zweiadrig
1 Stecker zweipolig

Abb. 21
Quelle: Foto selbst erstellt

Die Funktion des DK51 Board wurde in dem Kapitel 2.2 erörtert.

4.2 Das Modell des Mischautomaten

Nach Betätigung des Starttasters S0 werden nacheinander 4 Flüssigkeiten zugegeben und gemischt. Die Mischung erfolgt während des Zulaufs der Flüssigkeiten. Das Zulaufen der Flüssigkeiten wird über die Fühler B1 bis B4 kontrolliert. Nach Schließen des Ventils Y4 durch den Fühler B4 leuchtet die Startlampe H0 durch Dauerlicht noch 4 Sekunden, dann schaltet die Klarlampe H5 auf Dauerlicht. Nach Aufleuchten der Klarlampe H5 kann die Stoptaste S5 betätigt werden. Die Anlage kann dann abgeschaltet und durch den Starttaster S0 erneut in Betrieb genommen werden.[1]

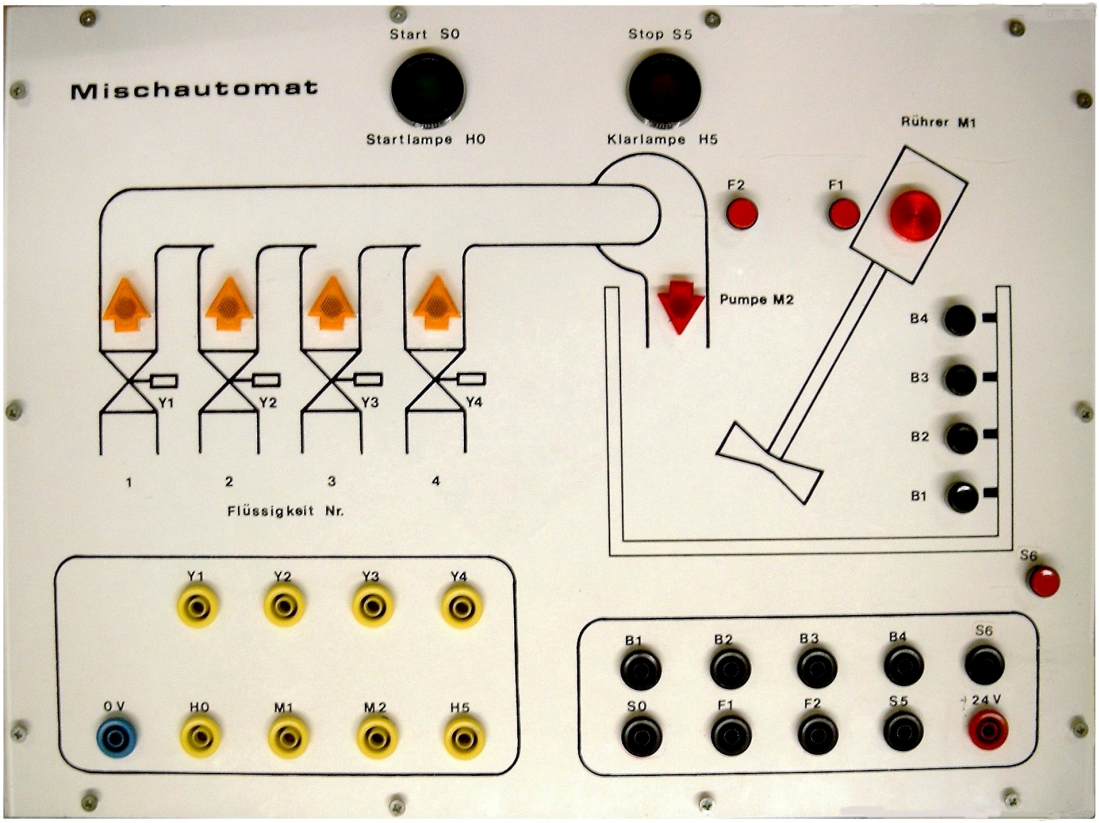
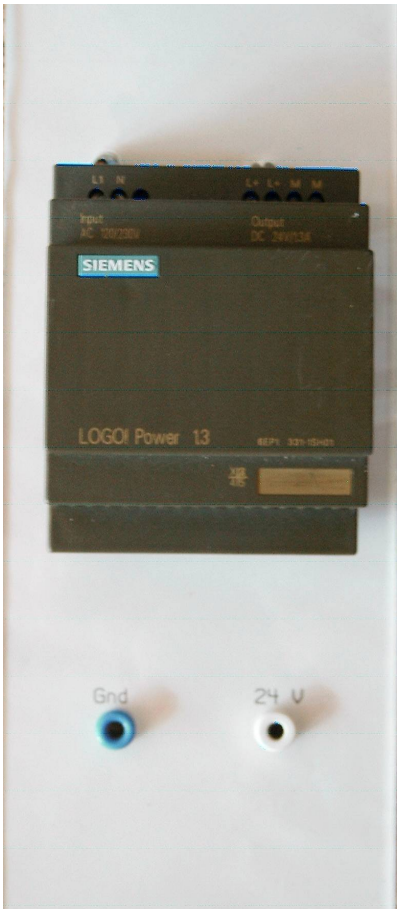


Abb. 22

Quelle: Foto selbst erstellt

[1] Quelle Schuleigene Beschreibung

4.3 Der Trafo



Der Siemens Transformator ist über eine Hutschiene auf der Grundplatte befestigt. Der Trafo wird mit 230V über eine dreiadrige Zuleitung mit Schukostecker versorgt. Auf der Rückseite der weißen Platte ist die Anschlussleitung mit einer Zugentlastung fixiert. Der Trafo hat eine Ausgangsspannung von 24 Volt DC (Gleichspannung), die an den Buchsen Gnd und 24V zur Verfügung stehen.

Stückliste

1 Grundplatte (weiß)
4 Gewindeschrauben
2 Buchsen
1 Stück Hutschiene
1 Zugentlastung

Abb. 23
Quelle: Foto selbst erstellt

4.4 Das Interface

Das Interface dient als Schnittstelle zwischen dem DK51 Board und dem SPS Modell. Die 24Volt Signale von dem SPS Modell sind mit dem Interface an den Buchsen zu verbinden (24V Din0 bis 24V Din7). Die Ausgänge vom DK51 Board werden mit den Buchsen (5V Din0 bis 5V Din7) verbunden. Als Spannungsversorgung dienen die 24Volt des Trafos (über die Buchsen 24V und Gnd) und die 5Volt vom DK51 Board (zweiadriges Kabel mit Stecker), welches direkt auf der selbst gelöteten Platine angeschlossen wird. Von den Buchsen aus, werden die Signale über ein 32 poliges Flachbandkabel auf die selbst angefertigte Lochraster Platine geführt. Das Platinen Layout ist mit dem Programm Lochmaster angefertigt wurden und befindet sich im Anhang (siehe Kapitel 10.1). Die erstellte elektronische Schaltung auf der Platine, ist in der Abb. 24 und 26 zu sehen (die Abb. 25 zeigt eine verbesserte Schaltung).

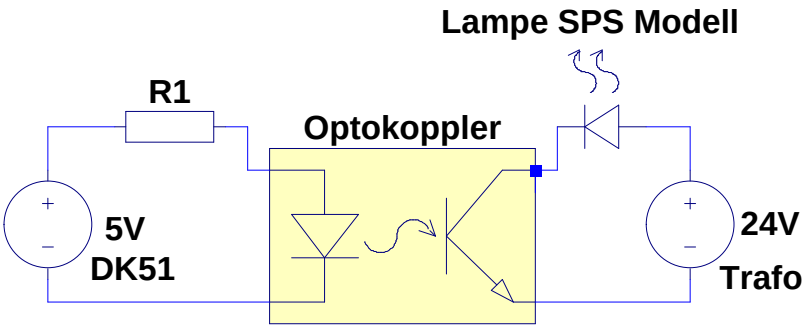


Abb. 24

Quelle: selbst erstellt mit SwCAD

Signalweg vom DK51 zum SPS Modell

Die zu verarbeitenden Signale von den Ausgängen des DK51 Boards sind jeweils über einen Vorwiderstand (R1) an die Anode von der (LED) des Optokoppler geschaltet. Optokoppler sind von der Funktion mit einem Relais zu vergleichen. Sie besitzen intern einen Fototransistor und eine Leuchtdiode (LED), die gegenüber angeordnet sind. Versorgt man die LED mit Strom leuchtet sie und macht den Fototransistor leitend. Der Vorteil ist, dass eine Galvanische Trennung der Signale erfolgt.

Der Widerstand begrenzt den Strom und senkt die Spannung für die Diode im Optokoppler herab. Die Betriebsspannung für die LED hängt vom Optokoppler ab, diese liegt z. B. bei 1,5V. Damit die Leuchtdiode mit den 1,5V versorgt wird müssen vom DK51 Ausgangssignal (5V TTL Pegel), 3,5V am Widerstand R1 abfallen. Der Diodenstrom ist auch unterschiedlich und muss aus dem jeweiligen Datenblatt entnommen werden. Bei einen optimalen Steuerstrom für die Leuchtdiode von z. B. 20mA, würde sich der Vorwiderstand zu (R=U:I) 3,5V:20mA=175Ω errechnen. Ein positives Signal bewirkt somit eine niederohmige Uce Strecke. Der durchgeschaltete Optokoppler schaltet die Spannungsquelle mit dem passenden Verbraucher durch (24V Trafo und Steuerlampe am SPS Modell). Problem bei der Schaltung ist, das die Uce Strecke nicht voll durchsteuert, ein Teil der Spannung vom Trafo fällt am Transistor ab. Somit stehen für den Verbraucher keine 24V mehr zur Verfügung, die Lampe leuchtet dunkler.

Lösungsvorschlag: Eine zweite Treiberstufe mit PNP Transistor (Abb. 25).

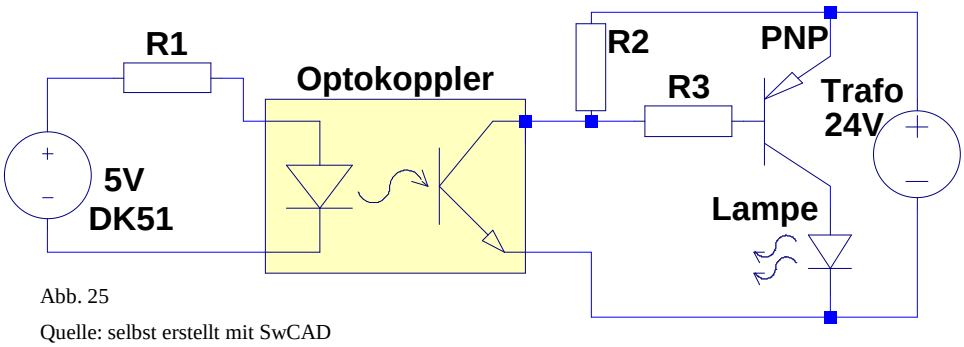


Abb. 25

Quelle: selbst erstellt mit SwCAD

Signalweg vom SPS Modell zum DK51

Die Funktion der Schaltung ist vergleichbar mit dem Signalweg vom DK51 Board zum SPS Modell. Eine Besonderheit in dieser Schaltung ist der Pulldown Widerstand R2. Die Eingänge vom Dk51 Board müssen mit einen 0 Signal (Masse) oder einen 1 Signal (5 Volt) angesteuert werden. Ist dies nicht der Fall, dann ist der Eingang nicht definiert und springt zwischen beiden Zuständen hin und her. Durch das Betätigen eines Tasters auf dem SPS Modell, wird ein 24V Signal an die LED gelegt. Der Transistor schaltet durch und der Eingang des DK51 Boards wird mit 5V versorgt (high Signal). Liegt kein Signal vom SPS Modell an, so wird der Eingang vom DK51 auch nicht angesteuert. Der Eingang bekommt jetzt über den Pulldown Widerstand ein Null Signal (low Signal) von der Masse.

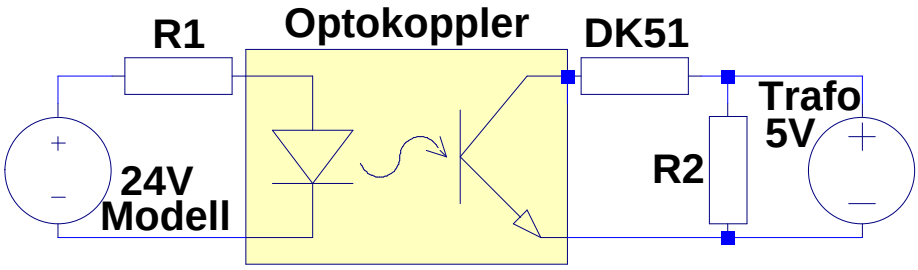
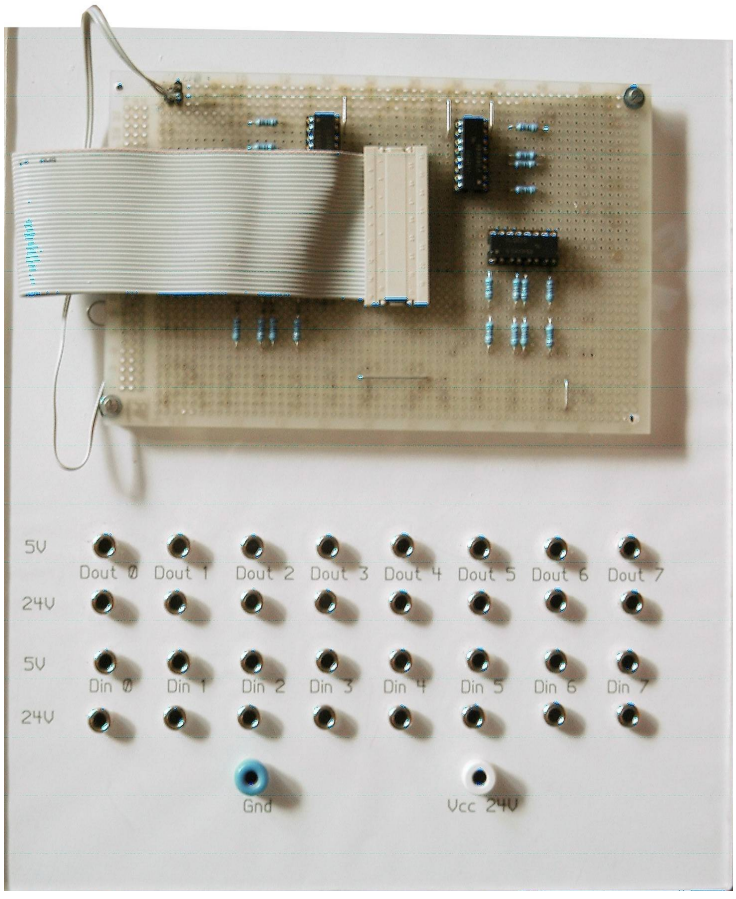


Abb. 26

Quelle: selbst erstellt mit SwCAD



Stückliste

1 Stück Platine

32 Buchsen in Silber

1 Buchse in Blau,

1 Buchse in Weiß

1 Stück Flachkabel

Steckerleiste

(16 polig)

24 Widerstände

4 Optokoppler (4fach)

4 Sockel Dil 16

1 Kabel

Abb. 27

Quelle: Foto selbst erstellt

5. Programmierung des DK51[1]

5.1 Arbeiten mit CoDeSys

Starten von CoDeSys

Das Programmiersystem CoDeSys über ***Start à Programme à 3S Software à CoDeSys V2.3 à CoDeSys V2.3*** starten.

Schreiben erster Programme

Das Mischermodell (im Kapitel 4.2 beschrieben) soll automatisiert werden. Am Beispiel des Programms zur Steuerung des Mischers wird das Programmieren in FUP (Funktionsplan) mit CoDeSys beschrieben. Das komplette Programm zur Steuerung des Mischers und die dazugehörige Aufgabenstellung befinden sich im Anhang unter Kapitel 10.2. Bevor mit dem Programmieren begonnen wird, ist die grundsätzliche Struktur des Programms festzulegen. Nach dem die Schrittkette angefertigt wurde, kann mit der Programmierung begonnen werden.

Anlegen eines neuen Projektes

Ein neues Projekt wird mit dem Befehl ***Datei à Neu*** erstellt.

Zielsystem Einstellungen

Für das erste Programm in der Combobox muss das Zielsystem entsprechend der Konfiguration ausgewählt werden, z. B. CoDeSys SP for Windows NT Realtime. Wenn die Installation des DK51 Bords erfolgreich abgeschlossen wurde, ist hier das ***DK51*** auszuwählen.

Der Baustein PLC_PRG

Im erscheinenden Dialog (neuer Baustein) ist die Sprache des Bausteins auszuwählen hier ***FUP***. Die Einstellungen Typ des Bausteins und Name des Bausteins auswählen. ***PLC_PRG*** ist ein besonderer Baustein, der im laufenden Betrieb zyklisch vom Laufzeitsystem aufgerufen und abgearbeitet wird. Dieser Baustein ist vergleichbar mit dem OB1 von Siemens Step7. In ihm müssen die Bausteine mit den Programmen gespeichert werden.

[1] Vgl. Erste Schritte mit CoDeSys

Baustein für Schritte einfügen

Den Ordner ***Bausteine*** (linker Bereich von CoDeSys) mit dem Kontextmenü (rechte Maustaste) den Befehl ***Objekt einfügen*** auswählen und einen weiteren Baustein einfügen**.** Der Name des Bausteins lautet ***Schritte***, Typ ***Programm*** auswählen und Sprache ***FUP.*** In diesem Baustein des Programms werden die Schritte geschrieben.

Deklaration einer Variablen

Für den ersten Schritt wird eine Variable benötigt (S1). Dazu müssen die drei Fragezeichen in dem ersten Netzwerk markiert werden. Dann wird der Name des Schalters ***S1*** eingeben und mit der Eingabetaste bestätigt. Im erscheinenden Dialog zur Variablendeklaration ist der Name ***S1*** und der Typ ***Bool*** bereits ausgewählt. Die Klasse auf ***VAR_GLOBAL*** (für globale Variablen) stellen. Mit *OK* wird die Variable im Objekt für die globalen Variablen eingetragen. Dieses Objekt befindet sich im Object Organizer unter ***Resourcen à Globale Variablen à Globale_Variablen***. Dort erscheint folgender Text:

```
VAR_GLOBAL
S1: BOOL;
END_VAR
```

Logische Bausteine einfügen

In der Grundstellung ist kein Schritt gesetzt (S1 bis S9 Negiert). Um ein logisches UND (AND) zu erstellen muss hinter der Variablen S1 mit der linken Maustaste geklickt werden, es erscheint ein kleines Quadrat. Über das Kontextmenü (rechte Maustaste) und den Befehl ***Baustein*** wird grundsätzlich der Baustein ***AND*** eingefügt, der Name ist markiert und kann nun durch den gewünschten Baustein ersetzt werden. Mit der Taste F2 wird ein Dialog geöffnet, in dem die verfügbaren Bausteine angezeigt werden. Für einen (ODER) Baustein muss statt ***AND*** einfach ***OR*** eingegeben werden. Ist die Bezeichnung nicht bekannt, kann über ***F2 à FUP Operatoren*** der richtige Baustein ausgewählt werden Um hinter einen bereits bestehenden Baustein einen weiteren einzufügen, muss mit der rechten Maustaste über das Kontextmenü ein neuer Baustein ausgewählt werden.

Zuweisen von Variablen

Eine Variable kann über das Kontextmenü mit dem Befehl ***Zuweisung*** zugewiesen werden. Es erscheinen am Ausgang des Bausteins drei Fragezeichen, hier muss die Variable eingegeben werden auf die zugewiesen werden soll.

Negationen

Um eine Negation einzufügen, muss der Ein- oder Ausgang markiert und über das Kontextmenü der Befehl ***Negation*** ausgewählt werden. Dieses bewirkt eine Invertierung des boolschen Signals und wird mit einem kleinen Kreis dargestellt.

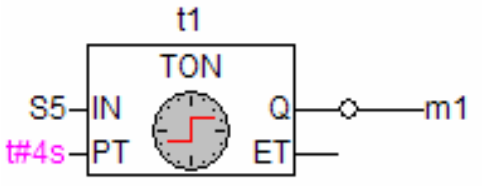


Abb. 28

Quelle: Screenshot selbst erstellt

Der Ausgang Q vom Baustein Ton wurde hier z. B. negiert. Für das Programm werden Anzugsverzögerte Zeitbausteine (TON) benötigt. Der Name des Bausteins kann frei gewählt werden, hier z. B. t1. Die Zeit kann wie abgebildet eingegeben werden.

Netzwerk einfügen

Um das Programm für Schritte zu schreiben muss nach Fertigstellung des ersten Netzwerks ein neues Netzwerk erstellt werden. Über den Menübefehl ***Einfügen à Netzwerk (danach)*** wird ein neues Netzwerk eingefügt. In den Netzwerken müssen die Bausteine für das Programm Schritte eingefügt werden. Danach muss ein neues Objekt eingefügt werden, zum Beispiel mit dem Namen Befehle und programmiert werden.

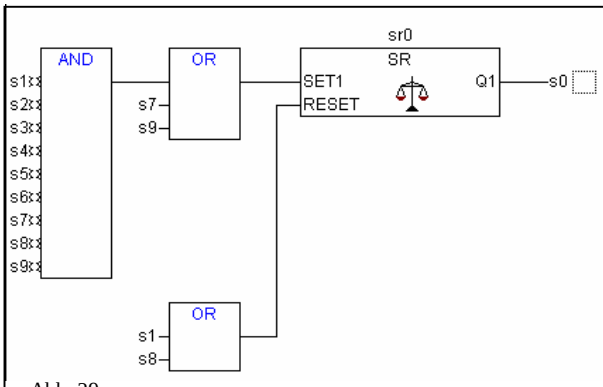


Abb. 29

Quelle: Screenshot selbst erstellt

In das Programm Schritte müssen die erforderlichen Bausteine für das Programm erstellt werden. Die Erstellung des Programms erfolgt aus der Schrittkette. Im Anschluss muss ein neues Objekt eingefügt werden, z. B. mit den Namen Befehle. Danach wird die Programmierung fortgesetzt.

Bausteine in den PLC_REG Baustein schreiben

In den Baustein ***PLC_PRG*** (vergleichbar mit dem OB1, in der S7 Software) müssen die erstellten Programme eingefügt werden. PLC_PRG wird pro Steuerungszyklus genau einmal aufgerufen. Im Kontextmenü unter dem Register ***Bausteine***, muss ***PLC_PRG*** mit der linken Maustaste (Doppelklick) geöffnet werden. Hier muss ein neues Netzwerk erstellt werden (rechte Maustaste und ***Netzwerk danach*** auswählen). In dem neuen Netzwerk muss ein neuer Baustein eingefügt werden. Der erstellte Baustein (AND) erscheint im Funktionsblock bereits markiert, mit ***<F2>*** wird der Eingabehilfe-Dialog angezeigt. Auf der linken Seite des Dialoges ***Definierte Programme*** auswählen. Die erstellten Programme werden jetzt unter ***Definierte Programme*** angezeigt und können ausgewählt werden. Hier z. B. den ***Baustein Schritte*** auswählen, für den Baustein Befehle muss dieses auch durchgeführt werden.

Vorsicht: Der Baustein **PLC_PRG** darf nicht gelöscht oder umbenannt werden (vorausgesetzt Sie verwenden keine Taskkonfiguration). PLC_PRG ist generell das Hauptprogramm in einem Single-Task Programm.

Projekt Übersetzen

Das Programm muss bevor es auf den Chip geladen wird in Maschinensprache übersetzt werden. Mit ***Projekt übersetzen*** wird das Projekt kompiliert. Der Übersetzungsvorgang ist grundsätzlich inkrementell, d.h. es werden nur die veränderten Bausteine neu übersetzt. Dazu im Menü ***Projekt à Alles übersetzen*** klicken oder mit der Funktionstaste ***<F11>*** den Befehl ***Alles übersetzen*** starten. In dem Meldungsfenster rechts unten sollte nach dem Übersetzungslauf „0 Fehler“ gemeldet werden. Falls Fehler im Programm vorhanden sind werden diese rot angezeigt. Mit einen Doppelklick auf die rote Fehlermeldung wird dann direkt auf den Fehler im Programm gesprungen. Der Fehler kann jetzt direkt bearbeitet werden, danach muss das Programm erneut übersetzt werden. Nachdem keine Fehler mehr vorhanden sind ist das Programm betriebsbereit. Der Übersetzungslauf, der mit **Projekt übersetzen** durchgeführt wird, erfolgt automatisch, wenn über ***Online à Einloggen*** in die Steuerung eingeloggt wird.

Damit der Chip aber angesteuert werden kann, muss zuerst eine Deklarierung der Ein- und Ausgänge erfolgen (siehe Kapitel 5.4). Für eine Simulation des Programms ist es sinnvoll eine Visualisierung anzulegen (siehe Kapitel 5.2).

5.2 Die Visualisierung

Visualisierung anlegen

Um eine Visualisierung anzulegen, muss im Object Organizer in der dritten Registerkarte von links, mit dem Namen ***Visualisierung,*** der Ordner ***Visualisierungen*** mit der rechten Maustaste markiert und im Menü ***Objekt einfügen*** ausgewählt werden. Jetzt wird dem Visualisierungsobjekt ein Namen gegeben, z. B. ***PLC_Visu*** (für Web Visualisierung).

Visualisierungselement zeichnen

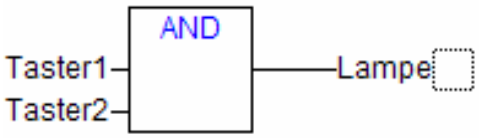


Abb. 30

Quelle: Screenshot selbst erstellt

Für den Baustein in der Abbildung, mit den Variablen (Taster1, Taster2 und Lampe) soll eine Visualisierung erstellt werden. Die Visualisierung soll wie in Abb.31 angefertigt werden.

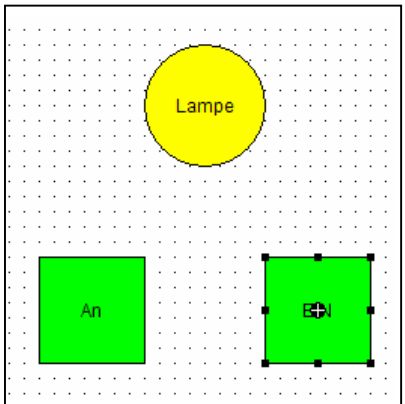


Abb. 31

Quelle: Screenshot selbst erstellt

Für den Taster1, in der Symbolleiste z. B. das Symbol ***Rechtecke*** mit der linken Maustaste auswählen und Taste wieder loslassen. Beim Platzieren die linke Maustaste gedrückt halten und ein Rechteck aufziehen. Hat es die gewünschte Form und Größe erreicht, kann die Maustaste losgelassen werden. Das Rechteck mit der linken Maustaste markieren um es dann evtl. an einen anderen Platz zu verschieben. Für den Taster2 das bereits erstellte Rechteck markieren, mit der rechte Maustaste das Menü öffnen und mit ***Bearbeiten à Kopieren*** und ***Bearbeiten à Einfügen*** eine Kopie vom ersten Rechteck erstellen (Vorteil, beide Rechtecke haben die gleiche Größe). Nach dem markieren des Rechtecks, kann es an der gewünschten Fläche platziert werden. Für die Lampe in der Symbolleiste z. B. den ***Kreis*** auswählen, und wie bei Taster1 an einen Platz einfügen und aufziehen.

Visualisierungselement konfigurieren

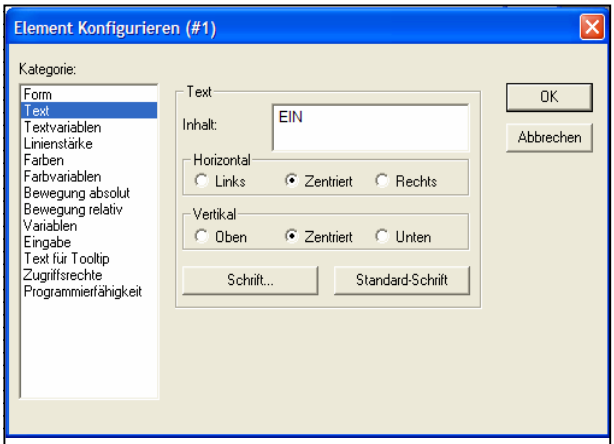


Abb. 32

Quelle: Screenshot selbst erstellt

Mit Doppelklick auf das erste Rechteck erscheint der Konfigurationsdialog für Taster1. In der Kategorie ***Text*** im Feld ***Inhalt*** den Namen oder die Funktion eingeben z. B. hier ***EIN.*** Die Kategorie ***Variablen*** auswählen und in das Feld ***Farbwechsel*** klicken und die Eingabehilfe mit der Funktionstaste <***F2***> aufrufen. Mit Doppelklick auf den Ordner ***Watch-Variablen*** in der rechten Hälfte des Dialogs erscheinen alle Variablen. Unter ***Globale_Variablen à Taster1*** auswählen. Anschließend die Kategorie ***Farben*** anwählen und auf die Schaltfläche ***Innen*** klicken eine Farbe (z. B. grün) auswählen. Danach eine weitere Farbe mit der Schaltfläche ***Innen*** im Bereich ***Alarmfarbe*** (z. B. rot) auswählen. In der Kategorie ***Eingabe*** den Punkt ***Variable tasten*** anklicken, und in dem Feld dahinter wieder die Variable mit <***F2***> auswählen. (Um ein Schalter zu definieren muss für Variable tasten ***Variable toggeln*** ausgewählt werden). Konfigurationsdialog verlassen. Damit zeigt das Rechteck im laufenden Betrieb die Farbe grün. Mit jedem Mausklick wird der Zustand gewechselt, und die Farbe wechselt von grün auf rot und umgekehrt (solange der Taster gedrückt wird). Die Einstellungen für das zweite Rechteck wiederholen. Für den Kreis die Variable toggeln oder tasten nicht auswählen, sonst wie beim ersten Rechteck konfigurieren.

Was bei einer Webvisualisierung zu beachten ist

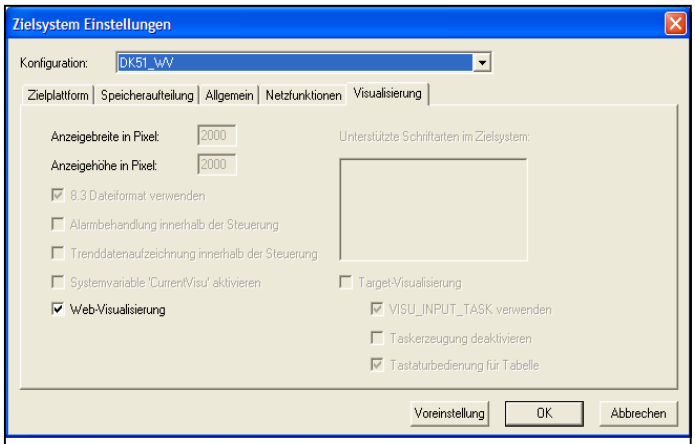


Abb. 33

Quelle: Screenshot selbst erstellt

Es gibt die Möglichkeit die Web Visualisierung über einen Explorer anzuschauen. Die Visualisierung muss plc Webvisu heißen. In dem Fenster ***Zielsystem Einstellungen à Visualisierung*** muss bei Web-Visualisierung ein Haken gesetzt werden. Jetzt kann im Explorer die IP Adresse des Chips / webvisu.htm eingegeben werden und dann z. B. http://192.168.1.113/webvisu.htm.

5.3 Zielsystem starten

Das Starten des Zielsystems in CoDeSys SP RTE ist nur mit Windows NT 4.0, Windows 2000 bzw. Windows XP möglich. Hier erscheint daraufhin das Icon für CoDeSys SP RTE in der Statusleiste. Mit der rechten Maustaste das Menü öffnen und ***Start System*** auswählen.

Einstellungen zum Verbindungsaufbau

Um die Verbindung mit dem Zielsystem aufzubauen sind einige Einstellungen vorzunehmen. ***Menübefehl Online à Kommunikationsparameter*** auswählen. In dem erscheinenden Dialog die Schaltfläche ***Neu***, um den Verbindungsaufbau zum Zielsystem zu konfigurieren. Ein Namen für die Verbindung eingeben und die entsprechende Verbindung zum Zielsystem auswählen. Für CoDeSys SP RTE Shared memory (Kernel). Verlassen des Dialoges mit ***OK.***

5.4 Die Deklaration der Ein- und Ausgänge

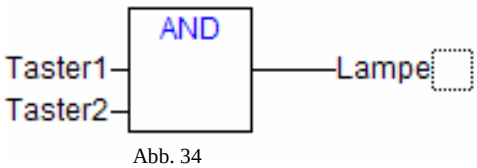


Abb. 34

Quelle: Screenshot selbst erstellt

Um die Ein- und Ausgänge zu zuweisen gibt es mehrere Möglichkeiten im Programm. Die Variablen Taster1, Taster2 und Lampe sollen deklariert werden.

Beim Erstellen einer Variablen z. B. Taster1 wird das Fenster Variablendeklaration angezeigt. Hier kann unmittelbar in das Feld Adresse der Ein- bzw. der Ausgang zugewiesen werden. Eingänge werden grundsätzlich mit ***%IX*** und Ausgänge mit ***%QX*** deklariert.
In der nächsten Abb. wird dem Taster1 der Eingang 0.1 zugewiesen.

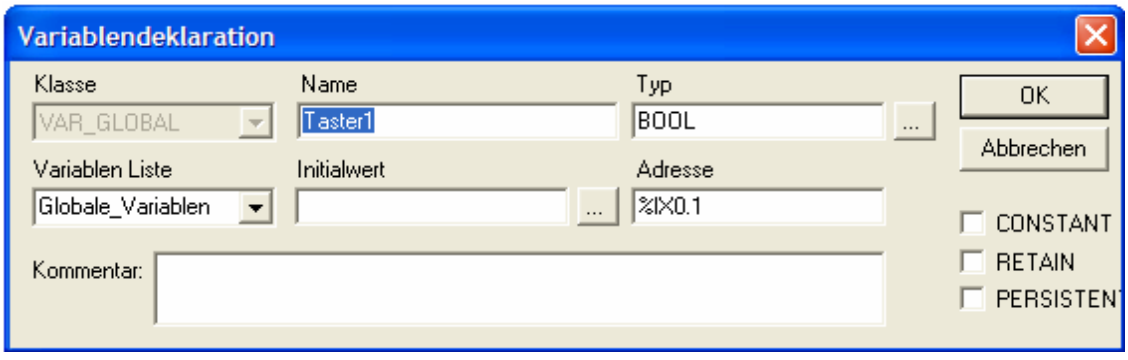


Abb. 35

Quelle: Screenshot selbst erstellt

Bei einer markierten Variablen ist es möglich unter ***Bearbeiten à Variablen Deklaration*** die Variable neu zu definieren.

Spätere Änderungen oder Zuweisungen sind jederzeit auch unter der Registerkarte ***Ressourcen à Ordner Globale Variablen à Globale Variablen*** möglich. Dabei ist zu beachten, dass vor dem Eingang bzw. Ausgang ein ***AT %*** geschrieben wird. Die Abbildung 37 zeigt die korrekte Schreibweise der Deklaration.

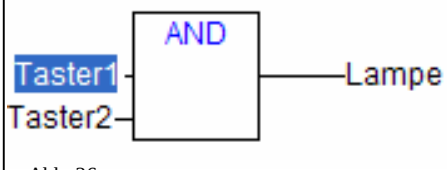


Abb. 36

Quelle: Screenshot selbst

```
VAR_GLOBAL
    Taster1 AT %IX0.1: BOOL;
    Taster2 AT %IX0.2: BOOL;
    Lampe AT %QX0.0: BOOL;

END_VAR
```

Abb. 37

Quelle: Screenshot selbst erstellt

5.5 Starten des Projektes

Mit dem Menübefehl ***Online à Einloggen*** wird ein Verbindungsaufbau zum Zielrechner hergestellt. Das Programm wird automatisch übertragen. **Mit *Online à Start*** wird das Programm auf dem Zielrechner gestartet (für eine Simulation, muss die Option ***Online Simulation*** aktiviert sein). Über die Visualisierung kann nun der Mischer bedient werden.

Weiteres Vorgehen

Es können weitere Applikationen erstellt und verwendet werden. Weitere Informationen enthalten die Online Hilfe und das Benutzerhandbuch zu CoDeSys.

6 Fazit

In diesem Bereich kann sicher noch viel in Form von Facharbeiten erarbeitet werden. Zum Beispiel können weitere binäre und analoge Ein- und Ausgänge an das DK51 System angeschlossen werden, um so komplexere Anwendungen zu steuern. Andere Projekte könnten mit Programmiersprachen wie z. B. C ++ erstellt werden. Laut Herrn Dipl. -Ing. Michael Linn der Firma Festo Didactic GmbH & Co. KG soll es sehr einfach sein, ein C Programm auf dem Chip auszuführen.

Weiterhin gibt es von der Schule die Überlegung das System im Unterricht einzusetzen. In dem System steckt viel Potential, da es so viele Möglichkeiten bietet. Man kann zum Beispiel Programme in IEC und C schreiben. Auf diese Weise können die beiden Programmiersprachen direkt miteinander verglichen werden. Im Rahmen der Facharbeit ist uns klar geworden, dass in Zukunft diese Technologien an Bedeutung gewinnen werden. Die Facharbeit hat verdeutlicht, dass die Einarbeitung in die beschriebene Technik sehr aufwendig ist. Die Dokumentation ist größtenteils nur auf Englisch verfügbar, dieses erschwert die Arbeit zusätzlich. Was noch bedacht werden sollte ist, dass das System sehr offen aufgebaut ist und es absichtlich oder unabsichtlich schnell zerstört werden kann. Auch die fehlenden Pulldown Widerstände und die nicht lieferbare Schnittstelle sprechen noch dagegen, das System in großen Stückzahlen für den Einsatz im Unterricht zu kaufen. Die Firma Beck ist jedoch dabei das System weiterzuentwickeln.
Die Schule wird dieses oder ein ähnliches System in naher Zukunft im Unterricht einsetzen.

7 Literaturverzeichnis

http://www.itwissen.info/de
http://www.beck-ipc.com
http://www.fibel.org/linux/lfo-0.6.0/node469.html
Erste Schritte mit CoDeSys
Getting Started V2.3
IPC World 02/2006
Computer & Automation 02/2006
Markt & Technik 16/04/2004
K-Magazin 01/2004
Elektronik Journal 04/2002
Quelle Schuleigene Beschreibung (für SPS Mischermodell)

8 Erklärung zur eigenständigen Anfertigung der Facharbeit

Hiermit erklären wir, dass wir die Facharbeit eigenständig angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel verwendet haben.

Wir versichern, dass wir alle Zitate und sinngemäßen Wiedergaben als solche kenntlich gemacht haben.

Hamburg, den 04.05.2006 _____

Unterschrift

Hamburg, den 04.05.2006 _____

Unterschrift

9 Einverständnisverklärung

Wir erklären uns damit einverstanden, dass unsere Daten für Presse- und Öffentlichkeitsarbeit verwendet werden.

Hamburg, den 04.05.2006 _____

Unterschrift

Hamburg, den 04.05.2006 _____

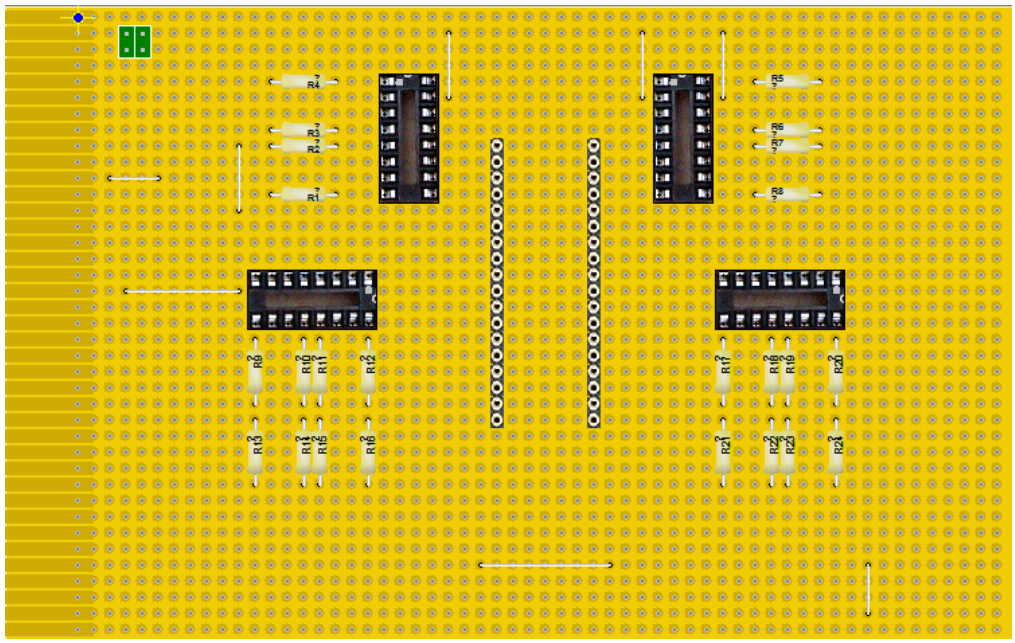
Unterschrift

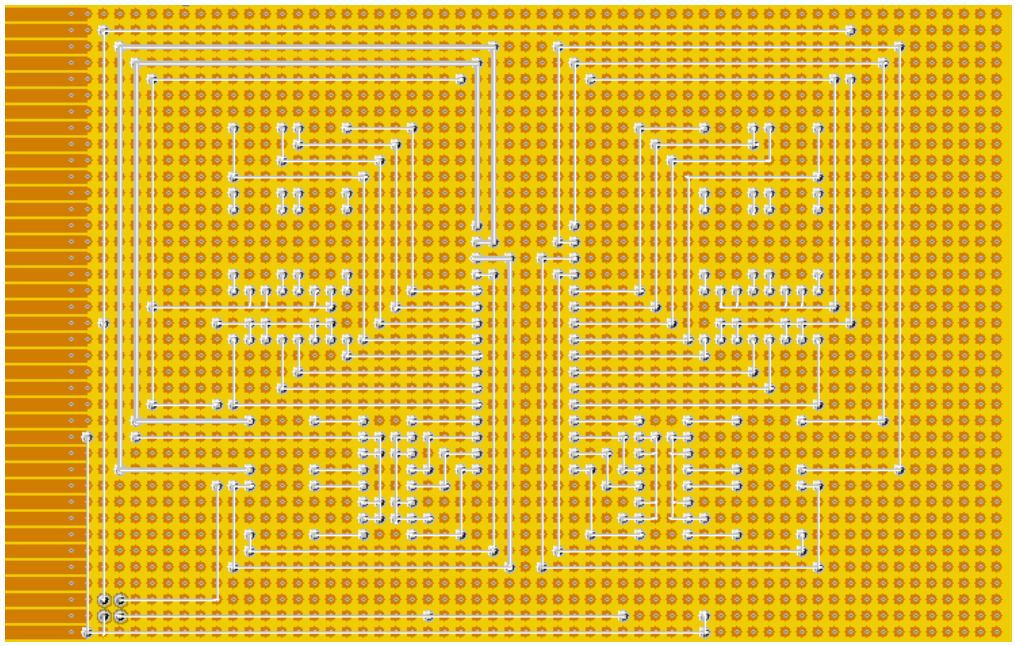
Anhang

- Schaltplan der selbst angefertigten Platine
- Mischerprogramm und Visualisierung
  - **PLC_PRG**
  - **Schritte**
  - **Befehle**
  - **Variablen**
  - **Visualisierung**

10 Anhang

10.1 Schaltplan der selbst angefertigten Platine

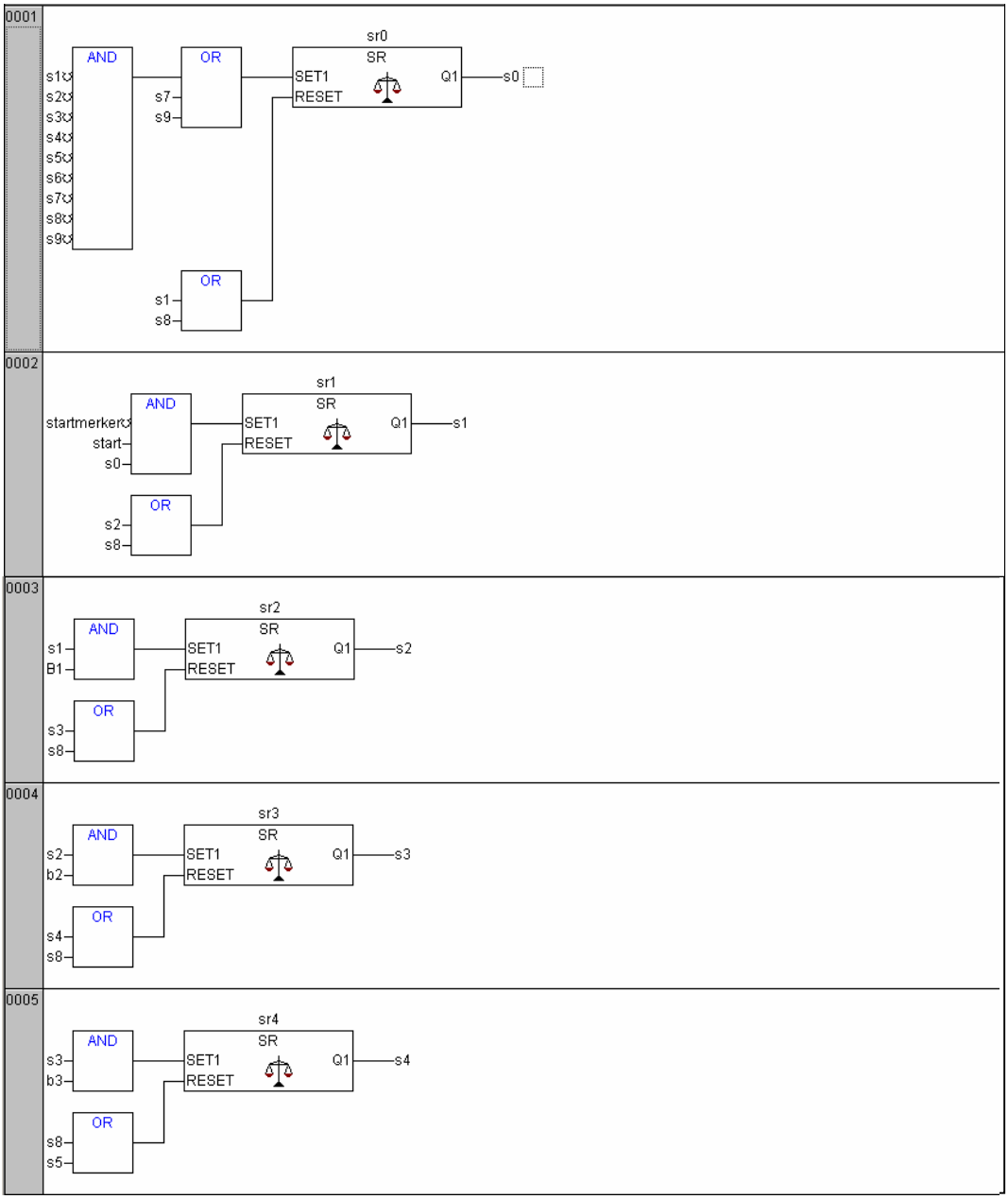




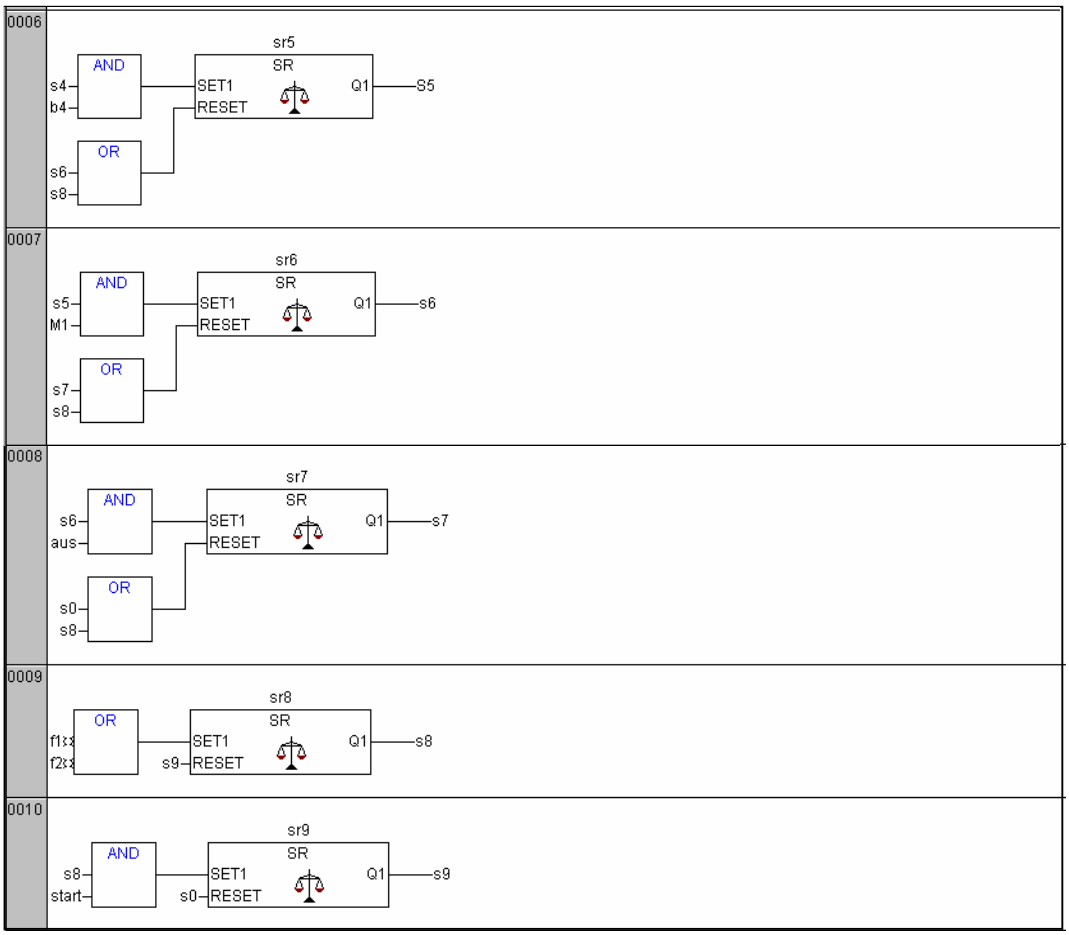
10.2 Mischerprogramm und Visualisierung

PLC_PRG

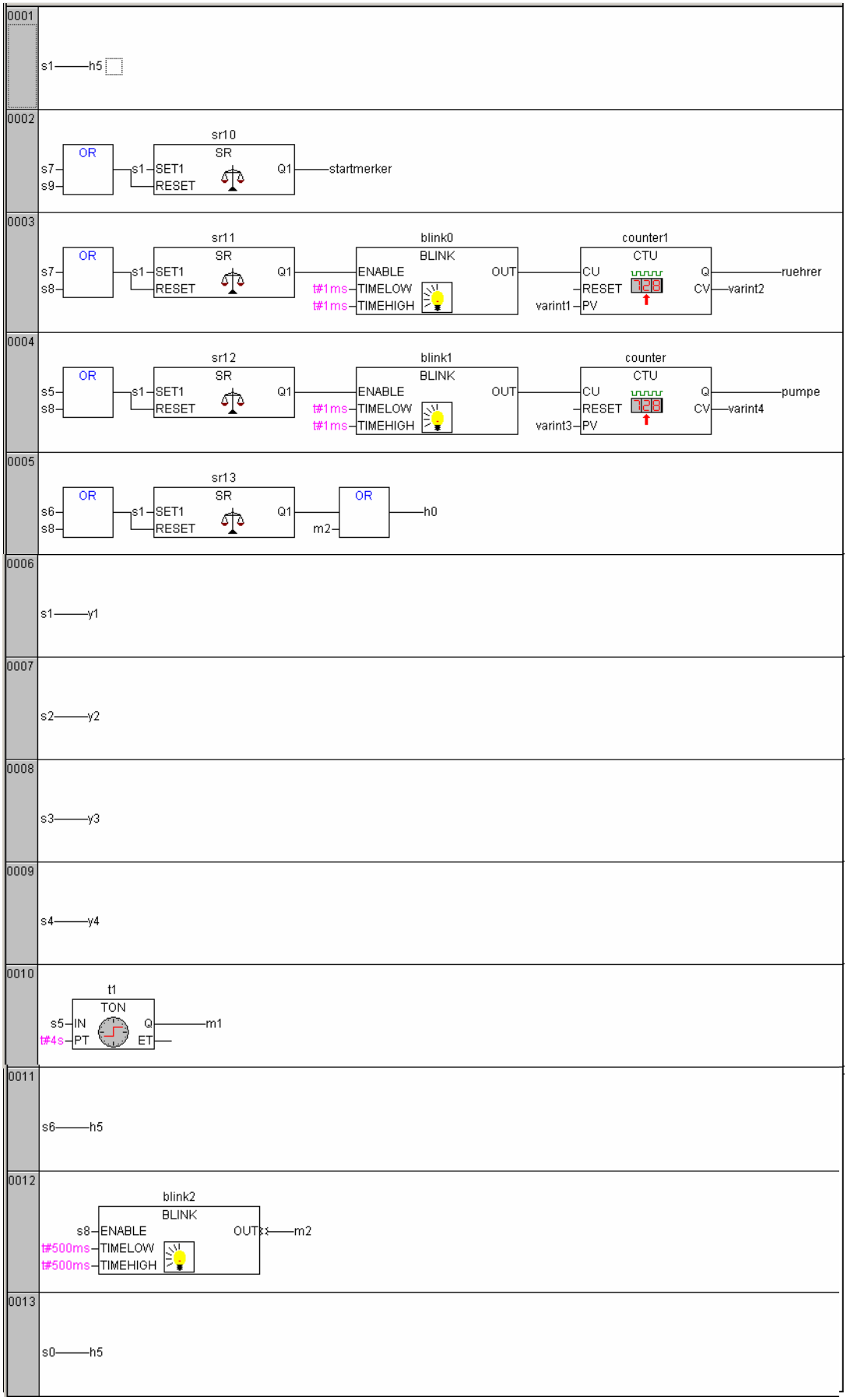
SCHRITTE



SCHRITTE


0006
sr5
AND
SR
s4
b4
SET1
RESET
Q1
S5
OR
s6
s8
0007
sr6
AND
SR
s5
M1
SET1
RESET
Q1
s6
OR
s7
s8
0008
sr7
AND
SR
s6
aus
SET1
RESET
Q1
s7
OR
s0
s8
0009
sr8
OR
SR
f1xx
f2xx
SET1
s9
RESET
Q1
s8
0010
sr9
AND
SR
s8
start
SET1
s0
RESET
Q1
s9

BEFEHLE


0001
s1
h5
0002
sr10
SR
OR
s7
s9
s1
SET1
RESET
Q1
startmerker
0003
sr11
blink0
counter1
BLINK
CTU
s7
s8
ENABLE
TIMELOW
TIMEHIGH
t#1ms
t#1ms
OUT
CU
RESET
PV
varint1
Q
CV
ruehrer
varint2
0004
sr12
blink1
counter
s5
s8
pumpe
varint4
varint3
0005
sr13
s6
m2
h0
0006
y1
0007
s2
y2
0008
s3
y3
0009
s4
y4
0010
t1
TON
s5
t#4s
IN
PT
Q
ET
m1
0011
s6
h5
0012
blink2
s8
t#500ms
t#500ms
OUT
m2
0013
s0
h5

VARIABLEN

```
VAR_GLOBAL
        s0: BOOL;
        s1: BOOL;
        s2: BOOL;
        s3: BOOL;
        s4: BOOL;
        s5: BOOL;
        s6: BOOL;
        s7: BOOL;
        s8: BOOL;
        s9: BOOL;
        startmerker: BOOL;
        M1: BOOL;
        h5: BOOL;
        ruehrer: BOOL;
        pumpe: BOOL;
        h0: BOOL;
        y1: BOOL;
        y2: BOOL;
        y3: BOOL;
        y4: BOOL;
        t1: TON;
        stoerung: BOOL;
        sr10: SR;
        sr11: SR;
        sr12: SR;
        sr13: SR;
        Start: BOOL;
        sr0: SR;
        sr1: SR;
        sr2: SR;
        sr3: SR;
        sr4: SR;
        sr5: SR;
        sr6: SR;
        sr7: SR;
        sr8: SR;
        sr9: SR;
        B1: BOOL;
        b2: BOOL;
        b3: BOOL;
        b4: BOOL;
        f1: BOOL;
        f2: BOOL;
        NOTaus: BOOL;
        aus: BOOL;
        quitiertaster: BOOL;
        BLINK: BLINK;
        varint1: INT;
    varint2:INT;
        counter: CTU;
        lampe: BOOL;
        varint3: INT;
        varint4: INT;
        blink2: BLINK;
        m2: BOOL;
        blink0: BLINK;
        blink1: BLINK;
END_VAR
```

Visualisierung